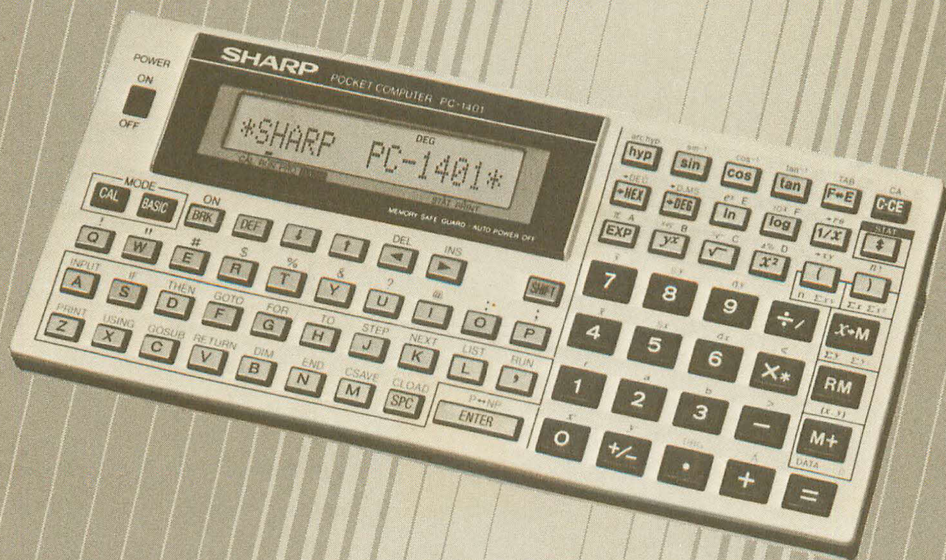
SHARP

ポケットコンピュータ

形名 PC-1401 PC-1402

取扱説明書

保証書付(巻末)
(WITH WARRANTY CARD)



信頼と保証の BMマーク
事務機械工業会 認定品

〈はじめに〉

このたびはシャープポケットコンピュータ〈PC－1401／PC－1402〉をお買いあげいただき、まことにありがとうございました。
この計算機は、くり返し計算や複雑な計算処理に威力を発揮するプログラム機能に加えて、電卓と同じように操作して計算をおこなうことができる関数電卓機能を持ったポケットコンピュータです。
本機を正しくお使いいただくために、この取扱説明書をよくお読みください。
なお、この取扱説明書は、「シャープサービス・お客様ご相談窓口一覧表」とともに、必ず保存してください。万一ご使用中にわからないことが生じたとき、きっとお役に立ちます。

（ご注意）

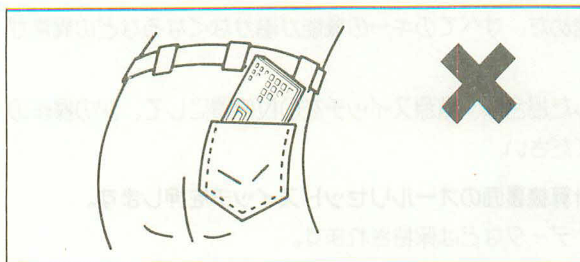
- この取扱説明書の内容は改良のため予告なく変更する場合があります。
- 本機は非常に複雑な機能および組合せを有する製品であり、出荷に際して取扱説明書を含め十分なチェックをして万全を期しておりますが、万一ご使用中ご不審な点・お気づきのことがありましたらもよりのシャープサービス・お客様ご相談窓口までご連絡ください。
  なお運用した結果生じる影響については責任を負いかねますので、あらかじめご了承ください。
- 本書に記載のプログラムを使用したことによる金銭上の損害および逸失利益または第三者からのいかなる請求についても当社はその責任を負いませんのであらかじめ、ご了承ください。

もくじ

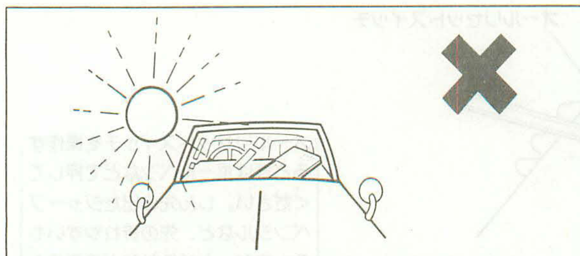
（ページ）

◆ おねがい………………………………………………………… 3
◆ 本機の主な特長……………………………………………… 5
◆ 第1章
　キーの操作と表示………………………………………… 7
◆ 第2章
　関数電卓として使用する方法…………………………… 15
◆ 第3章
　BASICに基づく計算……………………………………… 55
◆ 第4章
　プログラム…………………………………………………… 79
◆ 第5章
　その他………………………………………………………… 171
◆ アフターサービスについて………………………………… 192
◆ プログラム事例集………………………………………… 195
◆ エラーコードとエラー内容……………………………… 240
◆ キャラクタ・コード表…………………………………… 243
◆ BASICモードで使える関数・命令（省略形）一覧…245
◆ 索　引………………………………………………………… 255
保証書 ………………………………………………………… 259

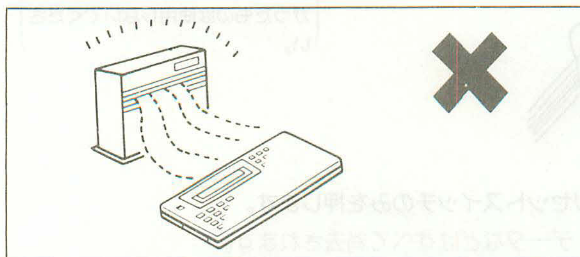
おねがい



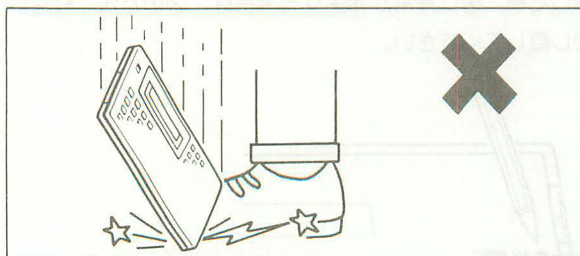
ズボンの後ポケットなどに入れないでください。座ったときなどに大きな力が加わり、特にガラスでできている液晶表示部が割れることがあります。



窓をしめきった、日の当たる自動車内に放置したり、直射日光の当たる場所に置いたりしないでください。高温により故障の原因になります。



暖ぼう器具の近くなど高温になる場所に置かないでください。



落としたり、ぶつけたりしないでください。



お手入れに揮発性の液体(シンナー、ベンジンなど)や、ぬれぞうきんなどは使用しないで、乾いた柔らかい布をご使用ください。

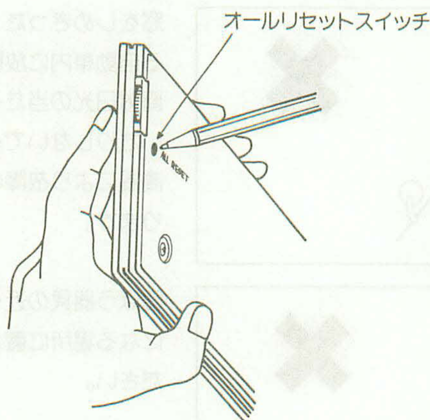
異常が発生した場合の処理について

本機をご使用中に強度の外来ノイズや強いショックをうけた場合などに、ごくまれに ON/BRK キーや電源スイッチも含めた、すべてのキーの機能が働かなくなるなどの異常が発生することがあります。

本機にこのような異常が発生した場合は、電源スイッチをON位置にして、次の操作のうち、どちらかをおこなってください。

① 任意のキーを押したまま計算機裏面のオールリセットスイッチを押します。

この場合は、プログラム、データなどは保持されます。

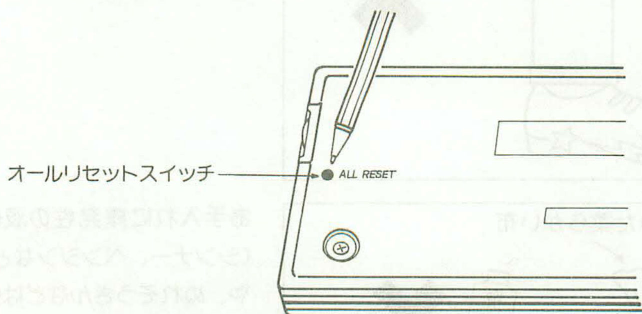


オールリセットスイッチを操作するときはボールペンなどで押してください。しん先の出たシャープペンシルなど、先の折れやすいもの、また、とくに針などの先のとがったものは使用しないでください。

② キーを押さずに、オールリセットスイッチのみを押します。

この場合は、プログラム、データなどはすべて消去されます。

①の方法で異常状態を解除した後、再び異常が発生する場合は、②の方法で解除し、もう一度プログラムを入力し直してください。



本機の主な特長

★初心者にも親しみやすいプログラム言語

本機は、プログラム言語にパーソナルコンピュータなどで広く使用されていますＢＡＳＩＣ（ベーシック）を用いています。

ＢＡＳＩＣは世界でもっとも多くの人が使用しているといわれるプログラム言語で、初めての人でも親しみやすく、少しの学習で容易にプログラムを組むことができます。

★ＰＣ－1245、ＰＣ－1250などで蓄積されたソフトが効率よく使えます。

本機は、発売以来好評をいただいておりますＰＣ－1250／ＰＣ－1251、ＰＣ－1245で開発、蓄えられたプログラムが使用できるように考慮されています。

これらのプログラムは表示に関連した部分などで、わずかな変更が必要なほかは、キーボードから入力、あるいはＣＥ－124、ＣＥ－126Ｐを介してテープから読み込ませるだけで、これまで蓄積したソフトが効率よく使えます。（くわしくは183ページをご参照ください。）

★大きなメモリー容量、しかもメモリー保護機能つき

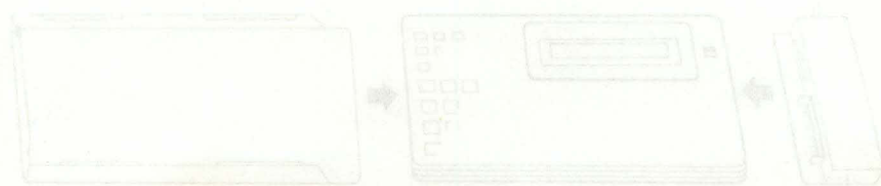
内部に40キロバイトＲＯＭ、4.2キロバイトＲＡＭ（ＰＣ－1402は10.2キロバイトＲＡＭ）を実装。

ＲＡＭに記憶されたプログラムやデータは、電源を切っても保護されていますので、使うたびに記憶させる必要がなく、いつでも、どこでもすぐに使用できます。

ＲＯＭ：Read　Only　Memory 読み出し専用メモリー
ＲＡＭ：Random　Access　Memory　読み書きができるメモリー
バイト：メモリーの大きさやプログラムの大きさを表わす場合などに用いる単位

★マニュアル計算は関数電卓機能でスピーディーに！

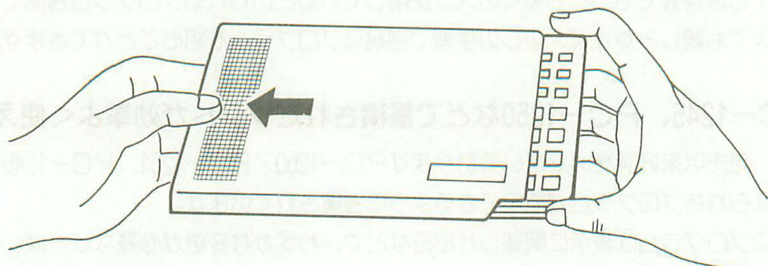
プログラムを組むほどでもない計算は、電卓と同じ感覚でキー操作をおこなえば、四則計算はもちろん、関数計算や統計計算までスピーディにおこなうことができます。



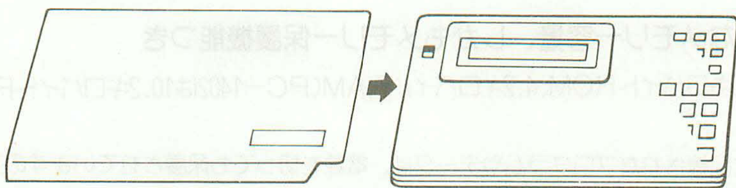
=ハードカバーの使いかた=

ハードカバーは衝撃に対して計算機を保護する役割を持っております。
計算機を使用しないとき、また、カバンに入れて持ち運ぶときなどには、ハードカバーを計算機に取りつけてください。

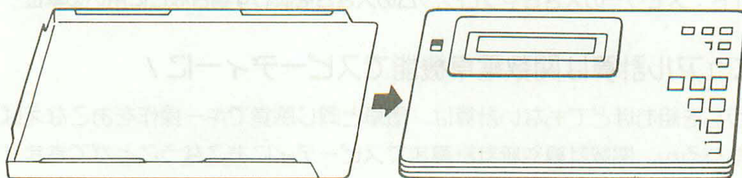
●ハードカバーの取りはずしかた



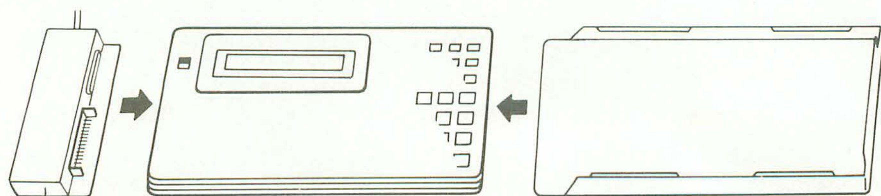
●計算機を使わないとき。



●計算機を使うとき。



●別売のカセットインターフェイスやプリンタを使うとき。



第1章

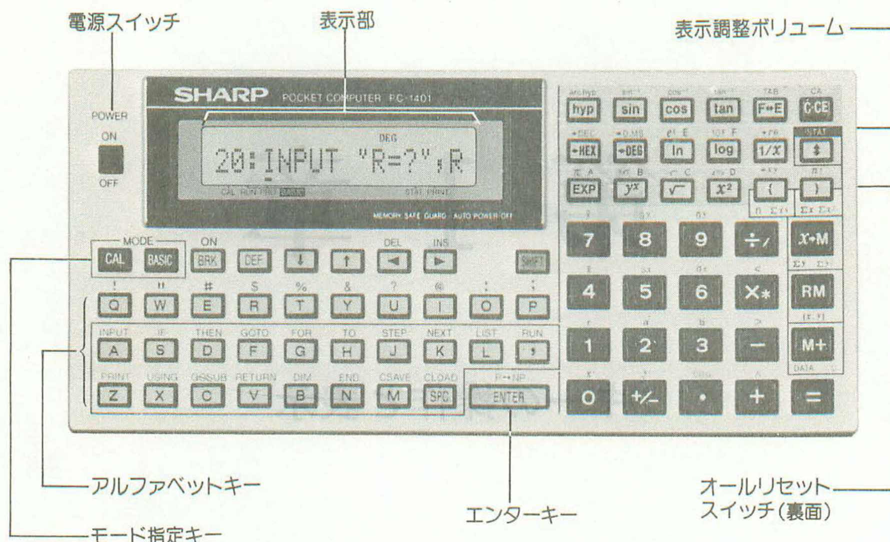
キーの操作と表示

■もくじ■

（ページ）

1．各部のなまえ……………………………………………8
2．スイッチ、キーの操作のしかた………………………9
3．モードについて…………………………………………14

1. 各部のなまえ



●写真はPC−1401です。

テンプレート

テンプレートは定義付けキーに定義付けしたプログラムの名称などを書き込み、キーボードにはめ込んで使用するものです。

例　定義付けしたプログラムの名称などを書き込んで使用（定義付けプログラム：121ページ参照）

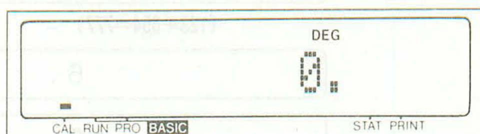
平均	面積	順位							
九九									

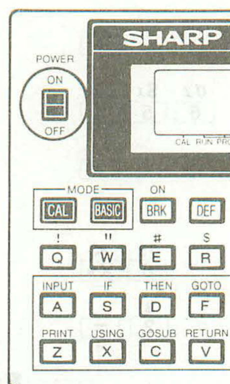
2．スイッチ、キーの操作のしかた

本機の上面には１個のスイッチと76個のキーがあります。そして、各キーにはいろいろな文字や数字、記号が書かれています。
ここでは、これらの操作のしかたについて簡単に説明します。

(1) 電源オン

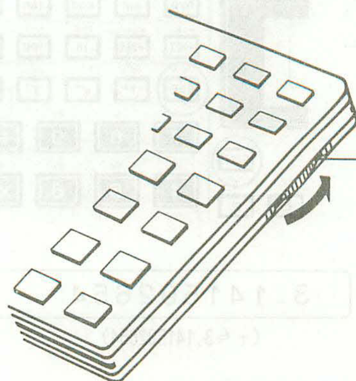
まず始めに、計算機の電源を入れましょう。
計算機の左端にある電源スイッチをOFF位置からON位置にしてください。電源が入って、次のような表示が出てきます。





●オートパワーオフ（AUTO POWER OFF）
本機はキー入力が可能なときに、約10分間新たなキー操作をおこないませんと、電池の消耗を少なくするため、自動的に電源が切れます。
この場合、ON/BRK キーを押せば電源が入ります。（169ページを参照）

〈表示の調整〉



コントラストボリューム

矢印の方向に回せば表示が濃くなり、逆方向に回せば薄くなります。
表示が見やすくなるように調整してください。

(2) キー操作のしかた

電源スイッチをONにして電源を入れたとき、この計算機は"電卓"として働くようになっています。

電卓はこれまでに何回か使ったことがありますね。

それでは簡単な計算をしてみましょう。次のようにキーを押して、表示を見てください。

キー操作	表示
[1] [2] [3]	123.
[+]	123.
[6] [5] [4]	654.
[=]	777. (123+654=777)
[6] [×*]	6.
[2] [=]	12. (6×2=12)

電卓として用いているのでイコールキーを押します。

キー操作	表示
[9]	9.
[√]	3. ($\sqrt{9}=3$)

どうですか、計算できましたか。もし、正しく計算できない場合は、電源スイッチをOFFにし、あらためてONにして計算をしてください。

それではつぎにπの値を呼び出してみましょう。

πは [EXP] キーの上側に薄茶色で書かれています。

このように薄茶色で書かれている機能は [SHIFT] キーに続いて押したときに働く機能です。

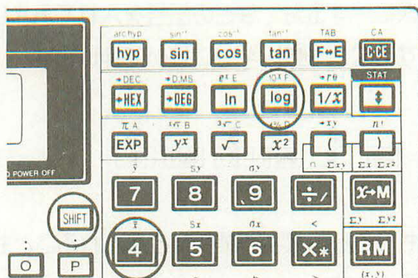
それでは [SHIFT] [EXP] と押してみてください。

キー操作	表示
[SHIFT] [EXP]	3.141592654 (π≒3.141592654)

πの値が呼び出されましたね。

こんどは10^4を計算してみましょう。10^4は10^xの機能で計算します。

この機能も薄茶色で書かれていますので SHIFT キーに続いて押します。



$\bar{x}$ 4 SHIFT 10^x log

10000.

$(10^4=10000)$

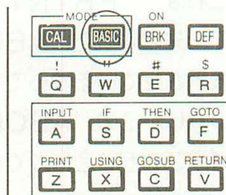
10^4 が計算できましたか？

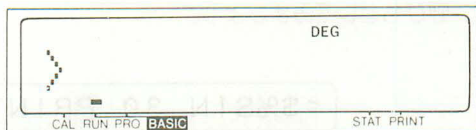
それでは次にアルファベットを入れてみましょう。

しかし、計算機を"電卓"として使っているときは、INPUT A 、IF S 、PRINT Z など、表示部の下にあるキーのほとんどは働きません。これらのキーはBASIC（ベーシック）の命令を入れたり、メッセージを入れたりするときに用いるキーです。"電卓"のときは使用できません。

それでは、これらのキーが働くようにしましょう。

左から2列目にある BASIC キーを押してください。表示が次のように変わります。





これでアルファベットキーなどが働くようになります。

それでは"ABCD"と入れてみましょう。

INPUT A DIM B GOSUB C THEN D

ABCD_

このように表示部の左端からABCDと入ります。

次に"#$%"などの記号を入れてみましょう。これらの記号は薄茶色で書かれていますので、SHIFT キーに続いて押します。

SHIFT E(#) SHIFT R($) SHIFT T(%)

ABCD#$%_

もちろん、数字や関数などを入れることもできます。たとえば"SIN30"と入れてみましょう。

sin

ABCD#$%SIN _

3 0

ABCD#$%SIN 3Ø_

BASICの命令も入れてみましょう。たとえば"PRINT"を入れます。

SHIFT Z(PRINT)

$%SIN 3Ø PRINT _

いま、"PRINT"を入れたときパッと表示が変わりましたね。
表示を見ていただければ、先に入れた内容の中から"ABCD#"がなくなっていることに気がつかれたと思います。
これは、"PRINT"を表示しようとすると、表示内容が16桁を越えてしまいますので、先に入れた"ABCD#"を表示の外に出して、あとから入れた"PRINT"を表示させているからです。
しかし、この"ABCD#"は消えてしまったわけではありません。DEL ◄ キーを押せば呼びもどすことができます。では、呼びもどしてみましょう。

DEL ◄

#$%SIN 3Ø PRINT

DEL ◄

ABCD#$%SIN 3Ø PR

このように DEL ◄ キーを押せば隠れていた内容が呼び出されます。

ここで、ちょっと DEL ◄ キーを押さえたままにしてみてください。
しばらくすると表示部で、ついたり消えたりしている ▓ マークが左端まで走っていきます。
左端まできて動かなくなったときは内容の一番先頭にきたことを示しています。

ABCD#$%SIN 3Ø PR

つぎに [INS ►] キーを押さえたままにしてみてください。
しばらくすると ▮ マークが右に走っていき、右端まで来たら、こんどは右に隠れている内容がつぎつぎに出て来ます。そして最後に ▮ マークが消えて _ マークが出ます。

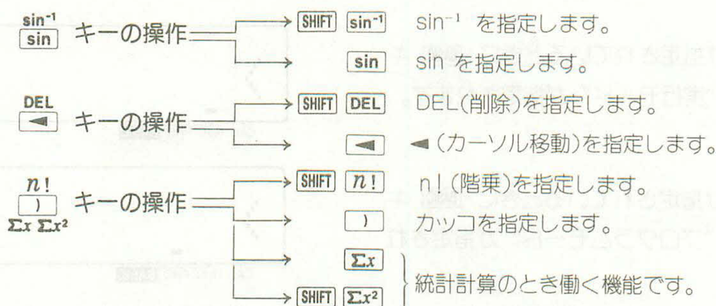
```
$%SIN 30 PRINT _
```

このように [DEL ◄] キーと [INS ►] キーで隠れている内容を見ることができます。

なお、▮ マークと _ マークは、カーソルといわれるもので、キーを押して入力される内容は、このカーソルで示される位置に入ります。
したがって、もし、キーを押しまちがえて、違う文字や記号が入ってしまったときは、[DEL ◄] キーでまちがった文字や記号の所にカーソルを移動させ、正しいキーを押せば訂正をすることができます。（くわしくは59ページで説明します。）

══キーの記載方法について══

これ以降の説明では、キー操作及びキーの機能を示す際、次のように記載します。



また、数字キーは、キーのワクを省いています。

[1] [2] [3] [・] [4] → 1 2 3 . 4

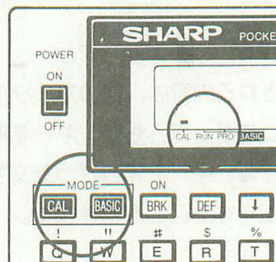
なお、説明上必要な場合などには、これまで通りの方法で示している場合もあります。

3．モードについて

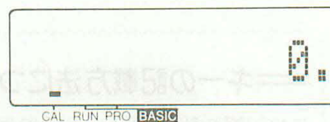
この計算機には、大きく分けて３つのモード（状態）があります。

１つは計算機を関数電卓のように使うことのできる“電卓モード”（CALモード）です。もう１つはプログラムを実行したり、BASICの命令などを使って手操作で計算を実行できる“実行モード”（RUNモード）です。そして最後はプログラムを計算機に覚えさせたり、訂正、修正したりできる“プログラムモード”（PROモード）です。

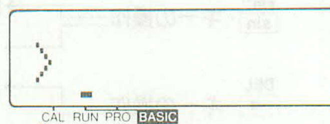
モードの切換えは CAL と BASIC の２個のモードキーでおこない、指定されているモードは、表示部左下に書かれている CAL RUN PRO の文字の上に ▬ マークをつけて示します。



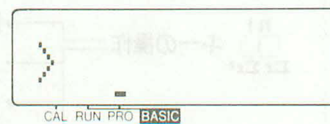
それでは一度電源スイッチをOFF位置にして、改めてON位置にしてください。このときは“電卓モード”がいつも指定されます。



電卓モードが指定されているときに BASIC キーを押せば“実行モード”が指定されます。



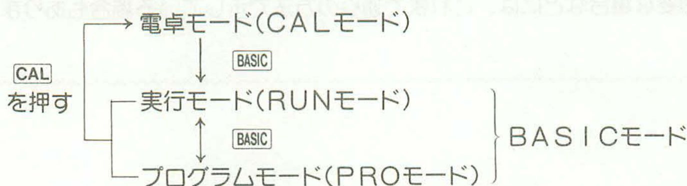
実行モードが指定されているときに BASIC キーを押せば“プログラムモード”が指定されます。



あとは BASIC キーを押せば交互に実行モードとプログラムモードが指定されます。

そして、 CAL キーを押すと、“電卓モード”にもどります。

〈モードの切換え〉



第２章

関数電卓として使用する方法

■もくじ■

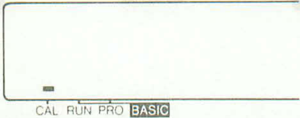
（ページ）
1．モードの指定……………………16
2．計算に使用できるキーと機能…17
3．表示の見かた……………………19
4．基本計算…………………………21
4－1）加減乗除算………………22
4－2）メモリー計算……………23
5．関数計算…………………………25
5－1）三角関数…………………25
5－2）逆三角関数………………26
5－3）双曲線関数………………27
5－4）逆双曲線関数……………27
5－5）対数関数…………………27
5－6）指数関数…………………28
5－7）２乗計算…………………28
5－8）平方根・立方根…………28
5－9）逆数計算…………………29
（ページ）
5－10）階　乗……………………29
5－11）角度変換…………………30
5－12）べき乗・べき乗根………31
5－13）座標変換…………………31
5－14）10進⇄16進変換、16進計算…… 33
5－15）パーセント計算…………36
5－16）変化率計算………………36
6．定数計算…………………………37
7．カッコ計算………………………39
8．小数部の桁指定のしかた………40
9．優先順位…………………………41
10．計算の保留………………………42
11．統計計算のしかた………………43
11－1）一変数統計計算…………44
11－2）二変数統計計算…………46
12．計算範囲…………………………51

1．モードの指定

計算機を関数電卓として使用する場合は、計算機のモード（状態）を〝電卓モード〞にします。

電卓モードは、電源スイッチをOFF位置からON位置にしたときに指定されるほか、CAL キーを押したときに指定されます。

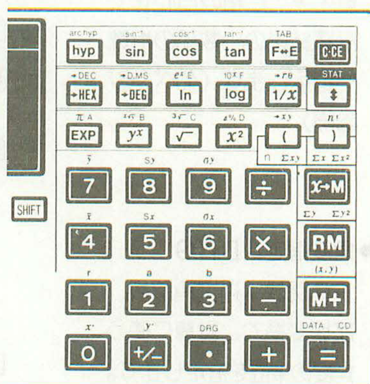
そして、表示部左下のCALと書かれているところに ▬ マークが表示されます。



もし、▬ マークがRUNやPROの位置に表示されているときは CAL キーを押してください。

2. 計算に使用できるキーと機能

電卓モードでは、右図に示しますキー及び機能を用いて計算をおこなうことができます。それでは始めに知っておいていただきたい機能を説明します。



● クリア　C・CE

置数のすぐあとや、メモリーの内容を呼び出したすぐあとでは、その数値のみをクリア（消去）します。

それ以外のときや、続けて2回押したときは、これまでの計算命令や数値をクリアします。

ただし、メモリーの内容などはクリアされません。

また、エラーの解除もおこないます。

キー操作	表示
123 [+] 456	456.
[C·CE]	0.
789 [=]	912.
	(123+789=912)
6 [×] 2 [+]	12.
[C·CE]	0.
6 [÷] 2 [+]	3.
5 [=]	8.
5 [÷] 0 [=]	0.（エラー表示 E）
[C·CE]	0.

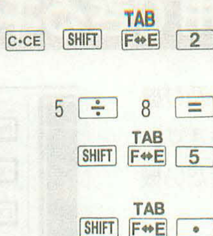
● 表示の切換え　F↔E

計算結果が固定小数点方式（普通の方式）で表示されているとき、浮動小数点方式（指数方式）で見ることができます。

キー操作	表示
23 [×] 1000 [=]	23000.
[F↔E]	2.3E 04
[F↔E]	23000.

● 小数部の桁数指定　TAB

数字キーと組み合わせて計算結果の小数部の桁数を指定します。(40ページ参照)



0.00
(2桁指定)

0.63

0.62500
(5桁指定)

0.625
(指定を解除)

● 角度指定　DRG

三角関数、逆三角関数、座標変換で、計算に使う角度の単位を指定します。

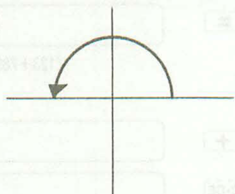
DEG
(ディグリー(度))



RAD
(ラディアン)

GRAD
(グラード)

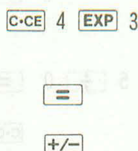
DEG
(ディグリー(度))



DEG：ディグリー(度)〔°〕
RAD：ラディアン　〔rad〕
GRAD：グラード　〔g〕

● 置数　0～9 . EXP +／－

EXP：数値を指数方式で入力するときに使用します。(表示では E になります。)

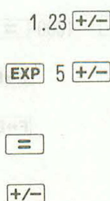


4.E 03
(4×10³)

4000.

−4000.

+／－：負数を入力するときに使用します。(表示数値の符号を反転させます。)



−1.23

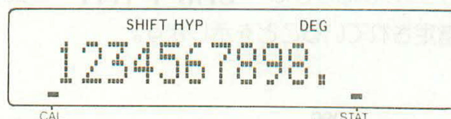
−1.23E−05
(−1.23×10⁻⁵)

−0.0000123

0.0000123

3. 表示の見かた

つぎに、電卓モードで使用する表示の形やシンボルなどについて説明します。



固定小数点表示
（普通の表示）

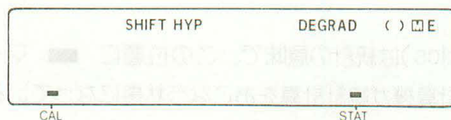


浮動小数点表示
（指数表示）

本機の表示部は16桁ありますが、電卓モードではこの表示部を左側12桁と右側４桁に分け、左側12桁で数値の符号と仮数部を表示し、右側４桁で指数部を表示します。
計算結果などを表示する場合、通常は固定小数点表示方式で表示しますが、0.000000001より小さいときや9999999999より大きいときなどには、自動的に指数表示に切換えて表示します。（負数のときは−0.000000001より大きいときや−9999999999より小さいときに指数表示になります。）

〈表示シンボル〉

つぎに表示シンボルや状態を示すマークについて説明します。



電卓モードでは上図に示しますシンボルやマークが用いられます。つぎにこれらの意味を述べます。

SHIFT：このシンボルは SHIFT キーを押したときに表示され、各キーの薄茶色で書かれている機能が指定されていることを示します。

HYP：このシンボルは arcHyp hyp キーを押したときに表示され、双曲線関数の機能が指定されていることを示します。

なお、 SHIFT arcHyp hyp と押したときは **SHIFT HYP** と表示され、逆双曲線関数の機能が指定されていることを示します。

DEG RAD GRAD

これらのシンボルは SHIFT DRG • と押すことにより、切換えて表示され、三角関数、逆三角関数、座標変換をおこなうときに使用される角度の単位を示します。

DEG：ディグリー（度）〔°〕

RAD：ラディアン〔rad〕

GRAD：グラード〔g〕

$(180^\circ = \pi \text{ [rad] } = 200^g)$

( )：このシンボルは →xy (キーを押してカッコ計算をはじめたときに表示され、カッコが使用されていることを示します。

M：このシンボルは電卓用メモリーに0以外の数値を入れたときに表示され、メモリーが使用されている（数値が記憶されている）ことを示します。

E：このシンボルはエラーが発生したときに表示されます。

エラーは C・CE キーを押せば解除されます。

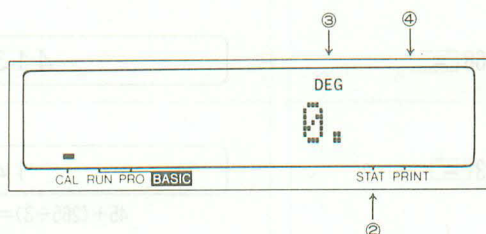
▬ STAT：SHIFT STAT ⇕ と押すことにより表示部下のＳＴＡＴと書かれているところに ▬ マークを表示させることができます。（消すときも SHIFT STAT ⇕ と押します。）

ＳＴＡＴ（Statistics）は統計の意味で、この位置に ▬ マークが表示されているときは、計算機が統計計算をおこなう状態になっていることを示します。

▬ CAL：ＣＡＬと書かれているところに ▬ マークが表示されているときは、計算機が電卓モードになっていることを示します。

4. 基本計算

ここでは本機を電卓として使用する場合の基本的な計算のしかたについて説明します。ただし、その前に計算機の状態を調えましょう。
まず、 CAL キーを押して "電卓モード" にしてください。
そして、 C·CE C·CE と押して、表示が次のようになっていることを確認してください。



もし、このようになっていない場合は次の説明を読んで、指定の変更をおこなってください。

① 0が2個以上表示されている。(例 0. 0)
小数部の桁数指定がおこなわれています。 SHIFT TAB • と押して指定を解除してください。

② STATと書かれている位置に ▬ マークが表示されている。
統計計算をおこなう状態になっています。 SHIFT STAT と押して ▬ マークを消してください。

③ **DEG**のかわりに**RAD**や**GRAD**が表示されている。
角度単位の指定です。三角関数、逆三角関数、座標変換をおこなうとき以外は、どのシンボルが表示されていてもかまいませんが、 SHIFT DRG の操作で切換えることができます。

④ M シンボルが表示されている。
電卓用メモリーに数値が記憶されています。このシンボルは新たにメモリーを使った計算をおこなうとき以外は表示されていてもかまいませんが C·CE x→M の操作で消すことができます。

⑤ 表示部上側に表示されるシンボルのうちに、③、④で説明のシンボル以外は C·CE キーで消すことができます。(表示部下側に表示される ▬ マークは、変りません。)

4－1）加減乗除算

例・操作	表示
〈 例 〉28＋42＋54＝ 〈操作〉28 [＋] 42 [＋] 54 [＝]	124.
〈 例 〉49.6−75.2＋32＝ 〈操作〉49.6 [−] 75.2 [＋] 32 [＝]	6.4
〈 例 〉1400×3−68＝ 〈操作〉1400 [×] 3 [−] 68 [＝]	4132.
〈 例 〉45＋285÷3＝ 〈操作〉45 [＋] 285 [÷] 3 [＝]	140. 45＋(285÷3)＝140
〈 例 〉(45＋285)÷3＝ 〈操作〉[(] 45 [＋] 285 [)] [÷] 3 [＝]	110.
〈 例 〉42×(−5)＋120＝ 〈操作〉42 [×] 5 [+/−] [＋] 120 [＝] ●負数を入力するときは「置数[+/−]」と押します。	−90.
〈 例 〉$\frac{18+6}{15-8}$＝ 〈操作〉[(] 18 [＋] 6 [)] [÷] [(] 15 [−] 8 [)] [＝] [＝]キー直前の[)]キーの操作は省略できます。	3.428571429
〈 例 〉2＋3×(−4＋5÷(12−4))＝ 〈操作〉2 [＋] 3 [×] [(] 4 [+/−] [＋] 5 [÷] [(] 12 [−] 4 [)] [)] [＝] 省略できます。	−8.125
〈 例 〉$(5\times10^{3})\div(4\times10^{-3})$＝ 〈操作〉5 [EXP] 3 [÷] 4 [EXP] 3 [+/−] [＝]	1250000.
〈 例 〉12000×24500×230＝ 〈操作〉12000 [×] 24500 [×] 230 [＝]	6.762E 10 (67620000000)

〈 例 〉1.2546÷123000÷42500= 〈操作〉1.2546 [÷] 123000 [÷] 42500 [=]	2.4E−10 (0.00000000024)

● 計算結果が±0.000000001～±9999999999および0以外になれば指数方式で表示されます。

4−2）メモリー計算

電卓用メモリーを使った計算をおこないます。メモリーのキーは [x→M] 、[RM] 、[M+] の3種類で、次のような働きになります。

[x→M]：メモリー内の数値を消してから、表示されている数値をメモリーに入れます。
[C·CE] [x→M] と押せばメモリー内の数値が消されます。

[RM]：メモリー内の数値を表示します。

[M+]：表示している数値、または計算をおこなったあと、その結果をメモリーに加算します。

ただし計算機が統計計算をおこなう状態になっているときは、メモリーを使うことはできません。

〈 例 〉
23+45+78=
−52−31+43=
+）64+73−12=
（合計）

〈操作〉	表示
[C·CE] [x→M]	0.
23 [+] 45 [+] 78 [M+]	M 146.
52 [+/−] [−] 31 [+] 43 [M+]	M −40.
64 [+] 73 [−] 12 [M+]	M 125.
[RM]	M 231.

● メモリーに0以外の数値が入っているときは [M] シンボルが表示されます。
計算を始める前に、このシンボルが表示されているときは [C·CE] [x→M] と押して、メモリー内の数値を消してください。

〈 例 〉　24×13＝
＋）56÷ 7＝
－）32× 4＝
（合計）

〈操作〉	表示
[C·CE] [x→M]	0.
24 [×] 13 [M+]	312.
56 [÷] 7 [M+]	8.
32 [×] 4 [=] [+/−] [M+]	−128.
[RM]	192.

● メモリーから減算をおこなうときは、この例のように計算をおこなったあと、[+/−] キーで符号を反転させてから [M+] キーを押してください。

〈 例 〉14－7×3＝
24÷(7×3)＝
(7×3)÷5＝

〈操作〉	表示
7 [×] 3 [=] [x→M]	21.
14 [−] [RM] [=]	−7.
24 [÷] [RM] [=]	1.142857143
[RM] [÷] 5 [=]	4.2

● この例では7×3の結果をメモリーに記憶させ、それを定数として計算をおこなっています。

5. 関数計算

● 三角関数、逆三角関数、座標変換をおこなうときは、 SHIFT DRG と押して計算に使用する角度の単位を指定してください。
ただし、角度単位を変更しない場合は、改めて指定する必要はありません。

5－1）三角関数（sin、cos、tan）

〈 例 〉 sin30°＝ 〈操作〉（ SHIFT DRG →DEGを表示させてください。） 30 sin	DEG Ø．5
〈 例 〉 $\cos\frac{\pi}{4}$〔rad〕＝ 〈操作〉（ SHIFT DRG →RADを表示させてください。） SHIFT π ÷ 4 ＝ cos	RAD Ø．7Ø71Ø6781 （$\frac{\pi}{4}$〔rad〕＝45°＝50^g）
〈 例 〉 tan150^g＝ 〈操作〉（SHIFT DRG→GRADを表示させてください。） 150 tan	GRAD －1． （150^g＝135°＝$\frac{3}{4}\pi$〔rad〕）
〈 例 〉（1－cos(2×35))÷2＝ 〈操作〉（SHIFT DRG→DEGを表示させてください。） C･CE (1 － (2 × 35) cos) ÷ 2 ＝	DEG Ø．328989928 ($\sin^2 35°$ ＝)

5−2）逆三角関数（$\sin^{-1}$、$\cos^{-1}$、$\tan^{-1}$）

● 逆三角関数の結果は次の範囲で表示されます。

$\theta=\sin^{-1}x$、$\theta=\tan^{-1}x$	$\theta=\cos^{-1}x$
DEG ：$-90\leqq\theta\leqq 90$〔°〕	DEG ：$0\leqq\theta\leqq 180$〔°〕
RAD ：$-\frac{\pi}{2}\leqq\theta\leqq\frac{\pi}{2}$〔rad〕	RAD ：$0\leqq\theta\leqq\pi$〔rad〕
GRAD：$-100\leqq\theta\leqq 100$〔g〕	GRAD：$0\leqq\theta\leqq 200$〔g〕

〈 例 〉 $\sin^{-1}0.5=$ 〔°〕

〈操作〉（SHIFT DRG →DEGを表示させてください。）

.5 SHIFT $\sin^{-1}$

DEG 3Ø.

（30°）

〈 例 〉 $\cos^{-1}-1=$ 〔rad〕

〈操作〉（SHIFT DRG →RADを表示させてください。）

1 +/− SHIFT $\cos^{-1}$

RAD 3.141592654

（3.141592654〔rad〕すなわちπ〔rad〕）

〈 例 〉 $\tan^{-1}1=$ 〔g〕

〈操作〉（SHIFT DRG →GRADを表示させてください。）

1 SHIFT $\tan^{-1}$

GRAD 5Ø.

（50^{g}）

〈 例 〉 $\cos^{-1}0.53-\sin^{-1}(40\div54)=$ 〔°〕

〈操作〉（SHIFT DRG →DEGを表示させてください。）

.53 SHIFT $\cos^{-1}$ − (40 ÷

54) SHIFT $\sin^{-1}$ =

DEG 1Ø.19999158

（約10.2°）

5−3）双曲線関数（sinh、cosh、tanh）

〈 例 〉cosh5+sinh5= 〈操作〉5 [hyp] [cos] [+] 5 [hyp] [sin] [=]	148.4131591
〈 例 〉$1+\tanh\frac{10}{2}=$ 〈操作〉1 [+] [(] 10 [÷] 2 [)] [hyp] [tan] [=]	1.999909204

5−4）逆双曲線関数（$\sinh^{-1}$、$\cosh^{-1}$、$\tanh^{-1}$）

〈 例 〉$\sinh^{-1}10=$ 〈操作〉10 [SHIFT] [archyp] [sin]	2.99822295
〈 例 〉$\cosh\left(3+\tanh^{-1}\frac{3}{4}\right)=$ 〈操作〉[C·CE] [(] 3 [+] [(] 3 [÷] 4 [)] [SHIFT] [archyp] [tan] [)] [hyp] [cos]	26.58007669

5−5）対数関数（log、ln）

〈 例 〉2×log5=　（$\log x = \log_{10} x$ ：10を底とする対数です。） 〈操作〉2 [×] 5 [log] [=]	1.397940009
〈 例 〉ln1.5+ln1.6=　（$\ln x = \log_e x$ ： eを底とする対数です。） 〈操作〉1.5 [ln] [+] 1.6 [ln] [=]	0.875468737

5−6）指数（逆対数）関数（10^x、e^x）

〈 例 〉$10^{2.5}=$ 〈操作〉2.5 [SHIFT] [10ˣ]	316.227766
〈 例 〉$\dfrac{2}{e^3+e^{-3}}=$ 〈操作〉2 [÷] [(] 3 [SHIFT] [eˣ] [+] 3 [+/−] [SHIFT] [eˣ] [)] [=]	0.099327927

5−7）2乗計算（x^2）

〈 例 〉$3^2+4^2=$ 〈操作〉3 [x²] [+] 4 [x²] [=]	25. (3×3+4×4=)

5−8）平方根・立方根（$\sqrt{\ }$、$\sqrt[3]{\ }$）

〈 例 〉$\sqrt{64}=$ 〈操作〉64 [√]	（$x\times x=64$　∴ $x=8$(あるいは−8)） 8.
〈 例 〉$\sqrt[3]{216}=$ 〈操作〉216 [SHIFT] [³√]	（$x\times x\times x=216$　∴ $x=6$） 6.
〈 例 〉$\sqrt[3]{\dfrac{250\times3}{\pi\times4}}=$ 〈操作〉[C·CE] [(] 250 [×] 3 [)] [÷] [(] [SHIFT] [π] [×] 4 [)] [=] [SHIFT] [³√]	3.907963209

5−9）逆数計算（$1/x$）

〈例〉$\frac{1}{4}+\frac{1}{8}=$ 〈操作〉4 [1/x] [+] 8 [1/x] [=]	Ø．375

5−10）階乗（$n!$）

$n!=n\times(n-1)\times(n-2)\cdots\cdots\times2\times1$

〈例〉5！＝ 〈操作〉5 [SHIFT] [n!]	5！＝5×4×3×2×1 12Ø．
〈例〉$_8P_3=\frac{8!}{(8-3)!}=$ 〈操作〉8 [SHIFT] [n!] [÷] [(] 8 [−] 3 [)] [SHIFT] [n!] [=]	順列の総数 $_nP_r=\frac{n!}{(n-r)!}$ 336．

5−11）角度変換（→DEG、→D.MS）

→DEG：数値の整数部を度、小数点以下1、2桁を分、3、4桁を秒と見なして10進数度単位に変換します。

→D.MS：数値を10進数度と見なして、度分秒に変換します。

○○．○○○○○○…
度　分　秒　端数(10進数)

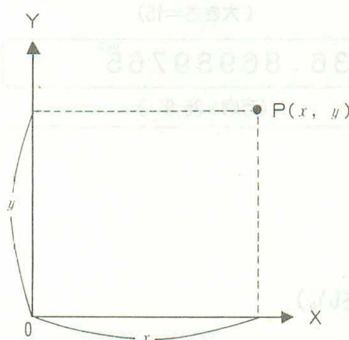
〈例〉12度39分18秒を10進数の度に変換します。	
〈操作〉12.3918 [→DEG]	12.655 (12.655度)
〈例〉124.628度を60進数の度分秒に変換します。	
〈操作〉124.628 [SHIFT] [→D.MS]	124.37408 (124度37分40秒8)
〈例〉　3時間30分45秒 +）6時間45分36秒 （　計　） 〈操作〉3.3045 [→DEG] [+] 6.4536 [→DEG] [=] [SHIFT] [→D.MS]	●角度変換機能を利用した時間計算です。 10.1621 (10時間16分21秒)

5−12）べき乗・べき乗根（y^x、$\sqrt[x]{y}$）

〈 例 〉3^4＝ 〈操作〉3 [y^x] 4 [＝]	81.
〈 例 〉$\sqrt[4]{81}$＝ 〈操作〉81 [SHIFT] [$\sqrt[x]{y}$] 4 [＝]	3.
〈 例 〉8^{-2}＝　$\left(\frac{1}{8^2}=\right)$ 〈操作〉8 [y^x] 2 [+/−] [＝]	0.015625
〈 例 〉$(12^3)^{\frac{1}{4}}$＝　$(\sqrt[4]{12^3}=)$ 〈操作〉12 [y^x] 3 [y^x] 4 [1/x] [＝]	6.447419591

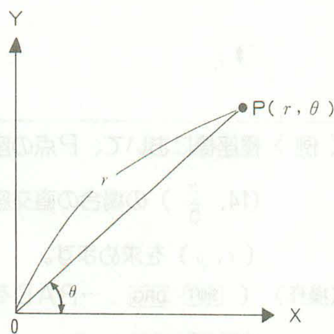
5−13）座標変換（→$r\theta$、→xy）

直交座標を極座標に（$x, y \to r, \theta$に）、あるいは極座標を直交座標に（$r, \theta \to x, y$に）変換します。



（直交座標）





（極座標）

〈→x, y〉

$x = r\cos\theta$　　$y = r\sin\theta$

〈→r, θ〉

$r = \sqrt{x^2 + y^2}$　　$\theta = \tan^{-1}\frac{y}{x}$

● θは次の範囲で得られます。

DEG：$0 \leqq |\theta| \leqq 180$

RAD：$0 \leqq |\theta| \leqq \pi$

GRAD：$0 \leqq |\theta| \leqq 200$

●計算をはじめる前に [SHIFT] [DRG] の操作で、計算する角度の単位を指定してください。

注）ほかの計算の途中で座標変換をおこないますと、それまで実行した計算およびカッコなどで保留されていた計算は消去されます。
また、統計モードが設定されているときは座標変換をおこなうことはできません。

〈 例 〉 直交座標において、P点の座標が(6, 4)の場合の極座標(r, θ)を求めます。

〈操作〉 ([SHIFT][DRG]→DEGを表示させてください。)

操作	表示
6[⇕]4	
[SHIFT][→rθ]	DEG 7.211102551 (r≒7.2)
[⇕]	DEG 33.69006753 (θ≒33.7°)

〈 例 〉 ベクトル $\dot{I}$=12+j9において、その大きさと方向(位相)を求めます。

〈操作〉 ([SHIFT][DRG]→DEGを表示させてください。)

操作	表示
12[⇕]9	
[SHIFT][→rθ]	DEG 15. (大きさ=15)
[⇕]	DEG 36.86989765 (方向≒36.9°)

〈 例 〉 極座標において、P点の座標が(14, $\frac{\pi}{5}$)の場合の直交座標(x, y)を求めます。

〈操作〉 ([SHIFT][DRG]→RADを表示させてください。)

操作	表示
[SHIFT][π][÷]5[=]	
[⇕]14[⇕]	
[SHIFT][→xy]	11.32623792 (x≒11.3)
[⇕]	8.228993532 (y≒8.2)

●この例では$\frac{\pi}{5}$の値(θの値)が先に入れられています。
したがって[⇕]14と押したときはrとθの値が逆に入っていることになります。このため、もう1回[⇕]キーを押してrとθの値を入れかえたあと、直交座標に変換しています。

5−14) 10進⇄16進変換、16進計算

(→HEX、→DEC)

10進⇄16進変換は、次のキーでおこないます。

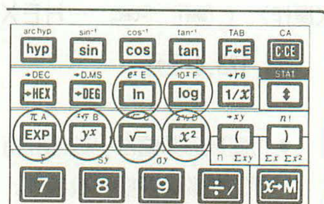
[→HEX]：10進数を16進数に変換します。
また、同時に16進モードにします。（HEXを表示させます。）

[SHIFT] [→DEC]：16進数を10進数に変換します。
また、同時に16進モードを解除します。（HEX表示を消します。）

16進法はコンピュータを扱う人びとの間でよく用いられる表記法で、0から9までの数字と、10進法の10から15までを表わす数字としてAからFをあてて表わします。

(16進数)		(10進数)
A	──────→	10
∫		∫
F	──────→	15

このAからFは、[→HEX] キーを押して表示部にHEXを表示させたあと、図で示しますキーを押せば入力することができます。



[→HEX] [EXP] [y^x] [$\sqrt{\ }$]
(π A　$\sqrt[x]{y}$ B　$\sqrt[3]{\ }$ C)

ABC.　HEX

なお、HEXの表示は、表示されている数値が16進数であることを示しているとともに、16進数の加減乗除算ができる状態（16進モード）になっていることを示します。
10進数を入力するときや、ほかの計算をするときは [SHIFT] [→DEC] と押してHEX表示を消してください。[C·CE] キーでは消せません。

〔10進→16進変換〕

〈 例 〉30を16進数に変換します。

〈操作〉30 [→HEX] | 1E. HEX

〈 例 〉－2を16進数に変換します。

〈操作〉([C•CE] [SHIFT] [→DEC] ……HEX表示を消します。)

2 [+/−] [→HEX] | FFFFFFFFFE. HEX

●負数の10進数を16進数に変換した場合、計算機内では"2の補数"計算がおこなわれ、結果は16の補数表示でおこなわれます。

注）HEXが表示されているとき、[+/−] キーで正数と負数を反転させた場合、負数は補数表示になります。

〈 例 〉123.4を16進数に変換します。

〈操作〉([SHIFT] [→DEC] …HEX表示を消します。)

123.4 [→HEX] | 7B. HEX

(123のみが16進数に変換されます。)

●10進数の数値が小数部を持っている場合、小数部は切り捨てられて、整数部のみが16進数に変換されます。

〔16進→10進変換〕

〈 例 〉2BCを10進数に変換します。

〈操作〉 [C•CE] [→HEX] 2BC | 2BC. HEX

[SHIFT] [→DEC] | 700.

〈 例 〉FFFFFFFF12を10進数に変換します。

〈操作〉[C•CE] [→HEX] FFFFFFFF12 | FFFFFFFF12. HEX

[SHIFT] [→DEC] | −238.

●FFFFFFFFFFからFDABF41C01までの16進数（補数）を10進数に変換した場合、その結果は負数（10進数）になります。

〔16進計算〕

16進数による計算は16進モードでおこないます。 [C·CE] [→HEX] と押して、表示部にHEXを表示させてください。

〈 例 〉A4+BA=

〈操作〉A4 [+] BA [=] 　　15E. HEX
(10進数では350)

〈 例 〉8×3=

〈操作〉8 [×] 3 [=] 　　18. HEX
(10進数では24)

〈 例 〉(12+D)×B=

〈操作〉 [C·CE] [(] 12 [+] D [)] [×] B [=] 　　155. HEX
(10進数では341)

〈 例 〉 43A−3CB=
+) A38−2FB=
(合計)

〈操作〉 [C·CE] [x→M]

43A [−] 3CB [M+] 　　6F. HEX

A38 [−] 2FB [M+] 　　73D. HEX

[RM] 　　7AC. HEX

16進計算は通常の計算とは異りますので、次の点を留意して計算をおこなってください。

- 16進計算では小数部を含む数値は扱いません。したがって [・] キーは入力しません。
- 連続的に計算をおこなう場合、途中結果が小数部を含む値になるときはエラーになります。

〈 例 〉B [÷] 3 [×] →エラー(E シンボルを表示)

ただし、最終結果が小数部を含む値になるときは、その小数部を切り捨てて整数部のみを表示します。

〈 例 〉B [÷] 3 [=] → 　　3. HEX

●16進モードでは [+/-] キーで補数表示がおこなわれます。(すでに補数表示になっているときは逆)

〈例〉AB [+/-] → FFFFFFFF55. HEX

[+/-] → AB. HEX

●16進モードでは、ほかの関数を使用することはできません。

●統計モード（STATのところに ▬ マークを表示しているとき）では16進計算、10進↔16進変換をおこなうことはできません。

5−15) パーセント計算

〈 例 〉200 g の60パーセントは?

〈操作〉200 [×] 60 [SHIFT] [Δ%]

120.

(120 g)

〈 例 〉120 g は500 g の何パーセント?

〈操作〉120 [÷] 500 [SHIFT] [Δ%]

24.

(24%)

5 −16) 変化率計算

〈 例 〉食塩250 g に食塩100 g を加えると、はじめの何パーセントになるでしょうか?

(式) $\frac{100+250}{250}\times 100=$

〈操作〉100 [+] 250 [SHIFT] [Δ%]

140.

(140%)

〈 例 〉1 秒間に25 ℓ 流れている水が、32 ℓ に増えました。さて、何パーセント増えたでしょう。

(式) $\frac{32-25}{25}\times 100=$

〈操作〉32 [−] 25 [SHIFT] [Δ%]

28.

(28%)

6. 定数計算

〈例〉12+42=
35+42=
68+42=

加算は加数が定数になります。

〈操作〉	表示
12 [+] 42 [=]	54.
35 [=]	77.
68 [=]	110.

└2式目以降は [+] 42の操作が省略できます。

〈例〉48−26=
32−26=
17−26=

減算は減数が定数になります。

〈操作〉	表示
48 [−] 26 [=]	22.
32 [=]	6.
17 [=]	−9

└2式目以降は [−] 26の操作が省略できます。

〈例〉742×450=
742×346=
742×265=

乗算は被乗数が定数になります。

〈操作〉	表示
742 [×] 450 [=]	333900.
346 [=]	256732.
265 [=]	196630.

└2式目以降は742 [×] の操作が省略できます。

〈例〉862÷8= 753÷8= 452÷8=	除算は除数が定数になります。
〈操作〉862 [÷] 8 [=]	107.75
753 [=]	94.125
452 [=]	56.5

└ 2式目以降は [÷] 8の操作が省略できます。

〈例〉12^4= 27^4=	y^x（べき乗）はxが定数になります。
〈操作〉12 [y^x] 4 [=]	20736.
27 [=]	531441.

└ 2式目以降は [y^x] 4の操作が省略できます。

〈例〉$\sqrt[5]{4700}$= $\sqrt[5]{3215}$=	$\sqrt[x]{y}$（べき乗根）はxが定数になります。
〈操作〉4700 [SHIFT] [$x\sqrt{y}$] 5 [=]	5.425247701
3215 [=]	5.028473844

└ 2式目以降は [SHIFT] [$x\sqrt{y}$] 5の操作が省略できます。

本機は、優先順位により計算が保留されます（41ページ参照）ので、連続計算をおこなった場合は計算機内部で最後におこなわれた計算と数値が、定数計算の計算命令と定数になります。

a＋b×c＝	a＋bc	＋bc	（定数加算）
a^b×c＝	a^b×c	a^b×	（定数乗算）
a×b÷c＝	ab÷c	÷c	（定数除算）
a－b÷c＝	$a-\frac{b}{c}$	$-\frac{b}{c}$	（定数減算）

7．カッコ計算

カッコはy^x、$\sqrt[x]{y}$、×、÷、＋、－の固有の優先順位通りの計算では不都合な場合、計算の順序を規定するために使用します。つまり“（”はその前にある計算の強制的な保留をおこない、（　）内を優先して計算をおこないます。

〈 例 〉72×(((56＋23)×21)−72÷(6−2))＝

〈 操作 〉72 [×] [(] [(] [(] → Ø.　( )

56 [+] 23 [)] [×] 21 [)] → 1659.　( )

[−] 72 [÷] [(] 6 [−] 2 [=] → 118152.

① カッコシンボルはカッコ計算がおこなわれていることを示しています。

② [=] キーや [M+] キーの直前にくる [)] キーの操作はすべて省略することができます。

8．小数部の桁指定のしかた

本機は計算結果の小数部の桁数を0～9桁の範囲で指定することができます。
小数部の桁数を指定することにより、小さくてあまり意味を持たない数字を表示しないようにしたり、計算結果の桁数をそろえたりすることができます。

指定は SHIFT TAB の操作に続いて 0 ～ 9 キーを押すことにより、それぞれの桁数が指定されます。

また、桁数の指定を解除するときは SHIFT TAB • と押します。

SHIFT TAB 0 →小数部桁を0桁に指定
　　　　～　　　　　～　　　　　　　　　指定された1桁下位が四捨五入で処理されます。
SHIFT TAB 9 →小数部桁を9桁に指定

SHIFT TAB • →小数部の桁指定が解除されます。
（上位桁から11桁目が四捨五入で処理されます。）

〈 例 〉 SHIFT TAB 9

.5 ÷ 9 =	→	Ø.Ø55555556	（小数点以下10桁目を四捨五入）
F⇔E	→	5.555555556E−Ø2	（仮数部の小数点以下10桁目を四捨五入）
SHIFT TAB 3	→	5.556E−Ø2	（仮数部の小数点以下4桁目を四捨五入）
F⇔E	→	Ø.Ø56	（小数点以下4桁目を四捨五入）
SHIFT TAB •	→	Ø.Ø5555555※	（有効数値の11桁目を四捨五入）

※計算機内では浮動小数点方式（指数方式）で、
$5.55555555555 \times 10^{-2}$
まで求め、仮数部11桁目を四捨五入して
$5.555555556 \times 10^{-2}$
とします。
そして、このあと固定小数点方式に変換して表示しますが、この例の場合は前にØが2つ付きますので、下位の桁は表示できません。したがって10桁目への切り上げがあっても、表示には表われていません。

9．優先順位

本機は計算の優先順位を判別する機能を備えていますので、基本的に数式通りのキー操作で計算をおこなうことができます。

〈優先順位〉

⑴ sin$x$、x^2などの関数

⑵ y^x、$\sqrt[x]{y}$

⑶ ×、÷

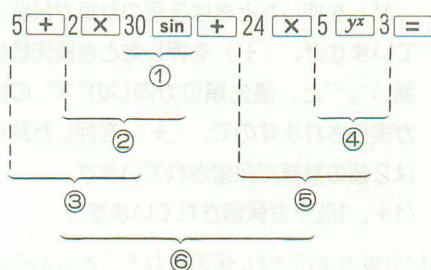
⑷ ＋、－

⑸ ＝、M＋、⊿%

（同一優先順位どうしの場合は順次計算が実行されます。）

なお、カッコが使用された場合はカッコ内の計算が優先されます。

例　5＋2×sin30＋24×5^3＝の計算のキー操作と計算順序



10．計算の保留

優先順位に従って計算を実行する場合や、カッコが使用された計算を実行する場合、すぐに実行できない計算を保留しておく必要があります。

本機では、計算(数値と計算命令)を最大８個まで保留しておくことができます。(８個を越えた場合はエラーになります。)

〈カッコを使用しない計算〉

例 ㋑ 1 [+] 2　　数値と計算命令を１個保留(1＋を保留)
　　　①

㋺ 1 [+] 2 [×] 3　　数値と計算命令を２個保留(1＋、2×を保留)
　　①　②

㋩ 1 [+] 2 [×] 3 [y^x] 4　　数値と計算命令を３個保留(1＋､2×､3y^xを保留)
　　①　②　③

㋥ 1 [+] 2 [×] 3 [y^x] 4 [÷] 5
　　①　②　③
　　　　②

[y^x] を押したときは３個の計算が保留されていますが、[÷] を押したとき優先順位の高い"y^x"と、優先順位が同じの"×" の計算が実行されますので、[÷] を押した時点では２個の計算が保留されています。
(1＋、162÷が保留されています。)

- y^x、$\sqrt[x]{y}$ を除く関数はキー操作後ただちに計算が実行され、保留されることはありません。
- カッコを使用しない計算においては、３個を越えて計算が保留されることはありません。

〈カッコを使用した計算〉

例 ㋑ 6 [+] 5 [×] 4 [y^x] [(] 3 [÷] 2　　数値と計算命令を４個保留
　　①　②　③　④

㋺ 6 [+] 5 [×] [(] 4 [−] 3 [÷] 2 [)]
　　①　②　③　④

[)] を押すとカッコ内の4－3÷2の計算が実行されるため、[)] を押した時点では２個の計算が保留されています。

- カッコは保留される計算が８個を越えない範囲で使用することができます。
  ただし、カッコ "("を連続して使用する場合は15個まで使用することができます。

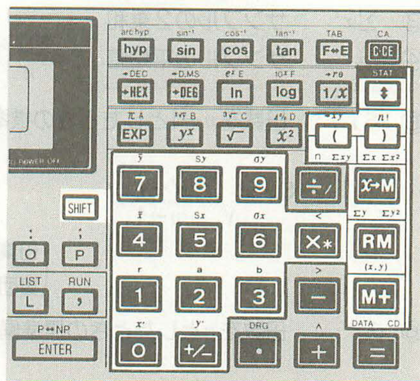
例　a × (((b − c × (((d ＋ e) × f) ÷ g………

カッコが、連続する場合は、それぞれ最大15個使用することができます。

11. 統計計算のしかた

統計計算をおこなう場合は SHIFT STAT と押して、統計モードを設定してください。

〈統計計算で主に使用するキー〉





SHIFT STAT と押して、この位置に ▬ マークを表示させてください。

統計モードを解除するときも SHIFT STAT と押します。

なお、統計量のうち次のものは、BASICモードのメモリー（固定変数領域）に記憶されており、統計モードを解除しても保持されていますので、BASICモードで使用することができます。

メモリー	Z	Y	X	W	V	U
統 計 量	n	Σx	Σx^2	Σxy	Σy	Σy^2

これらの統計量は、新たに統計モードを設定したときに消去されます。
したがって、新たに統計計算をおこなう場合は、統計モードを一度解除し、改めて設定してから計算をはじめてください。
そうしないと、前の統計量に新しい統計量が加わってしまいます。

- 統計モードが設定されているときは、メモリーやカッコを使用することはできません。
また、座標変換や、10進↔16進変換、16進計算をおこなうこともできません。

11−1）一変数統計計算

一変数統計計算では次の統計量を求めます。

① n　サンプル数

② $\sum x$　サンプルの総和

③ $\sum x^2$　サンプルの2乗和

④ $\bar{x}$　サンプルの平均値　$\bar{x}=\frac{\sum x}{n}$

⑤ Sx　母数をn−1としたときの標準偏差値

$$Sx=\sqrt{\frac{\sum x^2-n\bar{x}^2}{n-1}}$$

（母集団より抽出されたサンプルデータから、母集団の標準偏差を推定するときに使用します。）

⑥ σx　母数をnとしたときの標準偏差値

$$\sigma x=\sqrt{\frac{\sum x^2-n\bar{x}^2}{n}}$$

（母集団のすべてをサンプルデータとする場合、あるいはサンプルを母集団とみなして、その標準偏差を求める場合に使用します。）

データ入力は次の操作でおこないます。

① データを1つずつ入力する場合は

　データ [DATA]

　と押します。

② 同じデータが複数個ある場合は

　データ [×] 個数 [DATA]

　と押します。

（統計では、同じデータが何個かある場合は"度数"という言葉を使います。たとえば同じデータが3個ある場合は"度数3"のように表わします。）

〈 例 〉ある試験における点数を 、 ランダムに選び出した35人について見た場合、次のようになりました。
これより、平均値、標準偏差などを求めます。

№.	点　数	人　数	№.	点　数	人　数
1	30	1	5	70	8
2	40	1	6	80	9
3	50	4	7	90	5
4	60	5	8	100	2

〈 操作 〉 SHIFT STAT （統計モードを指定） — 0.

30 DATA 40 DATA — 2.
（サンプル数を表示）

50 × 4 DATA 60 × 5 DATA

70 × 8 DATA 80 × 9 DATA

90 × 5 DATA 100 × 2 DATA — 35.

SHIFT $\bar{x}$ （平均値 $\bar{x}$） — 71.42857143

SHIFT Sx （サンプルの標準偏差 Sx） — 16.47508942

SHIFT σx （母標準偏差 σx） — 16.23802542
（サンプルを母集団とみなしたときの標準偏差）

n （サンプル数 n） — 35.

Σx （総和 Σx） — 2500.

SHIFT Σx² （2乗和 Σx^2） — 187800.

SHIFT STAT （統計モードを解除） — 0.

● 途中結果として平均値や標準偏差などの統計量を求めたのち、データを入力すれば引き続き統計計算をおこなうことができます。また、統計量を求める順番は任意です。

〈データの訂正〉

違ったデータを入力してしまった場合は SHIFT CD で訂正します。

〈 例 〉 データ35が4個あるとき、まちがえて25 × 4 DATA と入力した場合

25 × 4 DATA　データを入れまちがえたときは
25 × 4 SHIFT CD　同じデータを入れて SHIFT CD と押し、
35 × 4 DATA　そのあと、正しいデータを入力します。

〈 例 〉 データ40が5個あるとき、まちがえて40 × 7 DATA と入力した場合

40 × 7 DATA　データの数を多く入れた場合は、
40 × 2 SHIFT CD　多い分を引きます。

11−2）二変数統計計算

二変数統計計算では次の統計量を求めます。

① n、$\sum x$、$\sum x^2$、Sx、σx は一変数統計計算と同じ

② $\sum y$　サンプル（y）の総和

③ $\sum y^2$ サンプル（y）の２乗和

④ $\sum xy$ サンプル(x, y)の積和

⑤ $\bar{y}$　サンプル(y)の平均値 $\bar{y}=\dfrac{\sum y}{n}$

⑥ Sy　サンプル(y)より求める、母数を(n−1)としたときの標準偏差値

$$Sy=\sqrt{\frac{\sum y^2-n\bar{y}^2}{n-1}}$$

⑦ σy　サンプル(y)より求める、母数を(n)としたときの標準偏差値

$$\sigma y=\sqrt{\frac{\sum y^2-n\bar{y}^2}{n}}$$

⑧ a　一次回帰線　y = a + b x の係数

$$a=\bar{y}-b\bar{x}$$

⑨ b　一次回帰線　y=a+bx の係数

$$b=\frac{Sxy}{Sxx}$$

⑩ r　相関係数

$$r=\frac{Sxy}{\sqrt{Sxx\cdot Syy}}$$

⑪ x'　推定値(yの値からxの値を推定)

$$x'=\frac{y-a}{b}$$

⑫ y'　推定値(xの値からyの値を推定)

$$y'=a+bx$$

$$\left[\begin{array}{l} Sxx=\sum x^2-\dfrac{(\sum x)^2}{n} \\ Syy=\sum y^2-\dfrac{(\sum y)^2}{n} \\ Sxy=\sum xy-\dfrac{\sum x\cdot\sum y}{n} \end{array}\right]$$

〈相関係数〉

相関係数(r)はxとyの2種類のデータがあるとき、たがいのつながり(相関関係)を数量的に表わすもので、次の範囲で得られます。

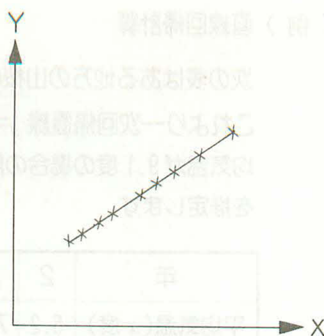
$-1 \leqq r \leqq 1$

この場合、rが1または−1のときは、相関図上の点がすべて、ある直線上にあることを示します。(図1)

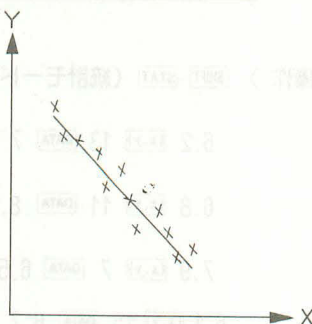
rが1または−1に近い場合は相関図上の点が、ある直線のまわりに密集していることを示します。(図2)

rが0に近い場合は相関図上の点が各方向に散らばり、なんらかの傾向を示すことが少ないことを示します。(図3)

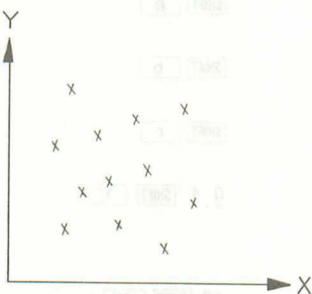
すなわち、rが1または−1に近いほど変量x、yの間に相関があることを示し、0に近いほど変量x、yの間に相関が少ないことを示します。



(図1)



(図2)



(図3)

なおr>0の場合は正の相関(xが大きくなればyも大きくなる：図1)関係を示し、r<0の場合は負の相関(xが大きくなればyは小さくなる：図2)関係を示します。

二変数統計計算では、次のようにxとyのデータを対にして入力します。

① データ(x) [(x,y)] データ(y) [DATA]

② 同じデータが何個かある場合は

データ(x) [(x,y)] データ(y) [×] 個数 [DATA]

と押します。

〈例〉直線回帰計算

次の表はある地方の山桜の開花日（4月）と同地3月の平均気温の表です。これより一次回帰直線 y=a+bx の係数a、bと相関係数rを求め、3月の平均気温が9.1度の場合の開花日、及び4月10日に開花した年の3月の平均気温を推定します。

年	2	3	4	5	6	7	8	9
平均気温(x度)	6.2	7.0	6.8	8.7	7.9	6.5	6.1	8.2
開花日 (y日)	13	9	11	5	7	12	15	7

〈操作〉 SHIFT STAT（統計モードを設定）　　0.（STAT）

6.2 (x,y) 13 DATA 7 (x,y) 9 DATA

6.8 (x,y) 11 DATA 8.7 (x,y) 5 DATA

7.9 (x,y) 7 DATA 6.5 (x,y) 12 DATA

6.1 (x,y) 15 DATA 8.2 (x,y) 7 DATA　　8.

SHIFT a　　34.44951017

SHIFT b　　−3.425018839

SHIFT r　　−0.969106837

9.1 SHIFT y′　　3.281838734

（推定：4月3日ごろ開花）

10 SHIFT x′　　7.13850385

（推定：3月の平均気温は約7.1°）

SHIFT STAT　（統計モードを解除）

〈例〉指数回帰計算

次のデータより $y = a \cdot e^{bx}$ の係数a、bおよび相関係数rを求め、x が12のときの y の値を推定します。

No.	1	2	3	4	5	6
x_i	2	7	9.2	4.3	5.1	8
y_i	0.6	4.02	8.3	1.21	2.7	5.1

計算のしかた

$y = a \cdot e^{bx}$ の両辺の対数を取りますと、

$\ln y = \ln a + bx$ ……①

となります。ここで、$\ln y$ をYとおき、$\ln a$ をAとおいて、①式を

$Y = A + bx$ ………②

の形にして、計算をおこないます。

なお、$Y = \ln y$ になっていますので、Yを入力するときは y の対数をとって入力します。

〈操作〉	表示
[SHIFT] [STAT]　（統計モードを設定）	0.　STAT
2 [(x,y)] 0.6 [ln] [DATA]	
7 [(x,y)] 4.02 [ln] [DATA]	
9.2 [(x,y)] 8.3 [ln] [DATA]	
4.3 [(x,y)] 1.21 [ln] [DATA]	
5.1 [(x,y)] 2.7 [ln] [DATA]	
8 [(x,y)] 5.1 [ln] [DATA]	6.
[SHIFT] [a] [SHIFT] [e^x]	0.307632838　（$a = e^A$）
[SHIFT] [b]	0.361879612
[SHIFT] [r]	0.983506276
12 [SHIFT] [y'] [SHIFT] [e^x]	23.65813575　（推定：y は約23.66）（$y = e^{(A+bx)}$）
[SHIFT] [STAT]　（統計モードを解除）	

＝ご注意＝

電卓モードで求めた統計量のうち、次のものはＢＡＳＩＣモードのメモリー（U~Z）に記憶されています。

メモリー	Z	Y	X	W	V	U
統計量	n	Σx	Σx^2	Σxy	Σy	Σy^2

これらの統計量を用いて計算をおこなう場合は、実行モード（ＲＵＮモード）でおこなってください。

たとえば、４個のデータ205,221,226,220を入力して、検定をおこなうときによく使用する偏差平方和を求める場合は次のように操作します。

〈式〉偏差平方和 $S^2=\Sigma(x-\bar{x})^2$

$$=\Sigma x^2-n\bar{x}^2$$

$$=\Sigma x^2-\frac{1}{n}(\Sigma x)^2$$

●まず電卓モードでデータを入力します。

CAL SHIFT STAT → `Ø.`

205 DATA 221 DATA

226 DATA 220 DATA → `4.`

●そして実行モード（ＲＵＮモード）で計算します。

BASIC → `>`

X − Y X* Y ÷/ Z → `X−Y*Y/Z_`

ENTER → `246.`

このように、上記の統計量を用いて計算をおこなう場合は、必らずＢＡＳＩＣモードでおこなってください。

12. 計算範囲

加減乗除算

被演算数、演算数、結果

±1×10^{-99} ～±9.999999999×10^{99} および0

関数計算

関数	計算範囲	備考
sin x cos x tan x	DEG : $\|x\|<1\times10^{10}$ RAD : $\|x\|<\frac{\pi}{180}\times10^{10}$ GRAD : $\|x\|<\frac{10}{9}\times10^{10}$ ただしtan x において次の場合は除く DEG : $\|x\|=90(2n-1)$ RAD : $\|x\|=\frac{\pi}{2}(2n-1)$ GRAD : $\|x\|=100(2n-1)$ （n は整数）	
$\sin^{-1} x$ $\cos^{-1} x$	$-1\leqq x\leqq1$	
$\tan^{-1} x$	$\|x\|<1\times10^{100}$	
ln x log x	$1\times10^{-99}\leqq x<1\times10^{100}$	ln x =$\log_e x$
e^x	$-1\times10^{100}<x\leqq230.2585092$	e≒2.718281828
10^x	$-1\times10^{100}<x<100$	
$\sqrt[3]{x}$	$\|x\|<1\times10^{100}$	
sinh x cosh x tanh x	$-227.9559242\leqq x\leqq230.2585092$	
$\sinh^{-1} x$	$\|x\|<1\times10^{50}$	
$\cosh^{-1} x$	$1\leqq x<1\times10^{50}$	
$\tanh^{-1} x$	$\|x\|<1$	
$\sqrt{x}$	$0\leqq x<1\times10^{100}$	
x^2	$\|x\|<1\times10^{50}$	
$\frac{1}{x}$	$\|x\|<1\times10^{100}$ $x\neq0$	

関数	計算範囲	備考
$n!$	$0\leqq n\leqq 69$（nは整数）	
D.MS→DEG	$\|x\|<1\times10^{100}$	
DEG→D.MS	$\|x\|<1\times10^{100}$	
y^x	● $y>0$のとき $-1\times10^{100}<x\log y<100$ ● $y=0$のとき $x>0$ ● $y<0$のとき xは整数または$\frac{1}{x}$が奇数 ただし$-1\times10^{100}<x\log\|y\|<100$	$y^x=10^{x\cdot\log y}$
$\sqrt[x]{y}$	● $y>0$のとき $-1\times10^{100}<\frac{1}{x}\log y<100,\ x\neq0$ ● $y=0$のとき $x>0$ ● $y<0$のとき xまたは$\frac{1}{x}$は整数（$x\neq0$） ただし$-1\times10^{100}<\frac{1}{x}\log\|y\|<100$	$\sqrt[x]{y}=10^{\frac{1}{x}\log y}$
DECI→HEX	$\|x\|\leqq 9999999999$	xは整数
HEX→DECI	$0\leqq x\leqq$ 2540BE3FF FDABF41C01 $\leqq x\leqq$ FFFFFFFFFF	xは16進数での整数
$x,\ y\rightarrow r,\ \theta$	$(x^2+y^2)<1\times10^{100}$ $\frac{y}{x}<1\times10^{100}$	$r=\sqrt{x^2+y^2}$ $\theta=\tan^{-1}\frac{y}{x}$
$r,\ \theta\rightarrow x,\ y$	$r<1\times10^{100}$ $\|r\sin\theta\|<1\times10^{100}$ $\|r\cos\theta\|<1\times10^{100}$	$x=r\cos\theta$ $y=r\sin\theta$ θの範囲は$\sin x$、$\cos x$のxと同じ

統計計算

統計量	計算範囲		備考
DATA CD	$\|x\|<1\times10^{50}$ $\|y\|<1\times10^{50}$ $\|\Sigma x\|<1\times10^{100}$ $\Sigma x^2<1\times10^{100}$	$\|\Sigma y\|<1\times10^{100}$ $\Sigma y^2<1\times10^{100}$ $\|\Sigma xy\|<1\times10^{100}$ $\|n\|<1\times10^{100}$	
$\bar{x}$	$n\neq0$		

統計量	計　算　範　囲	備　考
S_x	$n \neq 1$ $0 \leqq \frac{\sum x^2 - n\bar{x}^2}{n-1} < 1 \times 10^{100}$	
σx	$n \neq 0$ $0 \leqq \frac{\sum x^2 - n\bar{x}^2}{n} < 1 \times 10^{100}$	
$\bar{y}$	$n \neq 0$	
S_y	$n \neq 1$ $0 \leqq \frac{\sum y^2 - n\bar{y}^2}{n-1} < 1 \times 10^{100}$	
σy	$n \neq 0$ $0 \leqq \frac{\sum y^2 - n\bar{y}^2}{n} < 1 \times 10^{100}$	
r	$n \neq 0$ $0 < \left\| (\sum x^2 - n\bar{x}^2) \cdot (\sum y^2 - n\bar{y}^2) \right\| < 1 \times 10^{100}$ $\left\| \sum xy - \frac{\sum x \cdot \sum y}{n} \right\| < 1 \times 10^{100}$ $\left\| \frac{\sum xy - \frac{\sum x \cdot \sum y}{n}}{\sqrt{(\sum x^2 - n\bar{x}^2) \cdot (\sum y^2 - n\bar{y}^2)}} \right\| < 1 \times 10^{100}$	
b	$n \neq 0$ $0 < \left\| \sum x^2 - n\bar{x}^2 \right\| < 1 \times 10^{100}$ $\left\| \sum xy - \frac{\sum x \cdot \sum y}{n} \right\| < 1 \times 10^{100}$ $\left\| \frac{\sum xy - \frac{\sum x \cdot \sum y}{n}}{\sum x^2 - n\bar{x}^2} \right\| < 1 \times 10^{100}$	
a	bと同一でかつ$\left\| \bar{y} - \mathrm{b}\bar{x} \right\| < 1 \times 10^{100}$	
y'	$\left\| \mathrm{a} + \mathrm{b}x \right\| < 1 \times 10^{100}$	
x'	$\left\| \frac{y - \mathrm{a}}{\mathrm{b}} \right\| < 1 \times 10^{100}$	

計算の誤差は原則として、10桁目に±１の誤差となります。(指数表示の場合は仮数部表示の最下位桁に±１となります。)
ただし、関数の特異点および変曲点の近傍では誤差が累計されて大きくなります。
また、連続計算をおこなった場合はそれぞれの誤差が累積されて大きくなります。
(y^x、$\sqrt[x]{y}$ のように、計算機内で連続計算をおこなっている場合も同様です。)

第３章

ＢＡＳＩＣに基づく計算
（マニュアル計算）

■もくじ■

(ページ)
1. マニュアル計算とは……………… 56
2. マニュアル計算のしかた………… 57
3. キー操作の訂正…………………… 59
4. プレイバック機能………………… 61
5. 連続計算機能……………………… 62
6. ラストアンサー機能……………… 63
7. 数式の長さ………………………… 64

(ページ)
8 計算結果の表示方法…………… 65
9 メモリーの使いかた…………… 66
10. データメモリーと変数………… 68
10−1)変数の名前……………… 68
11. 関数計算………………………… 70
12. ダイレクトアンサー機能……… 76
13. 計算の優先順位………………… 77

１．マニュアル計算とは

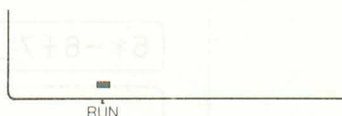
プログラムとして、計算の処理手順を計算機に記憶させて、自動的に計算させるのではなく、１つ１つの計算を１回１回手操作によって計算することをマニュアル計算と呼びます。もちろん、電卓モードでの計算はすべて手操作でおこないますので、マニュアル計算ですが、本書ではＢＡＳＩＣモード（実行モード、プログラムモード）でおこなう手操作による計算をマニュアル計算と呼びます。

本章では、プログラムを説明する前にＢＡＳＩＣに基づいた計算のしかたや、機能について説明します。

プログラムを作成する前に必らず知っておいていただきたい計算式の作りかたや関数、メモリーの使いかた、入れまちがえたときの訂正のしかたなど、基本的でかつ重要な事項を説明していますので、よく読んで覚えてください。

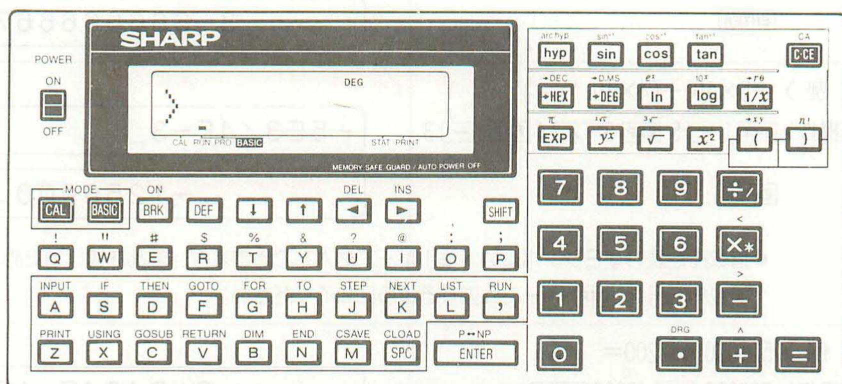
2. マニュアル計算のしかた

マニュアル計算は実行モード（RUNモード）でおこないますので、BASIC キーを押して実行モードにしてください。



BASIC キーを押して、RUNの上に ▬ マークを表示させてください。

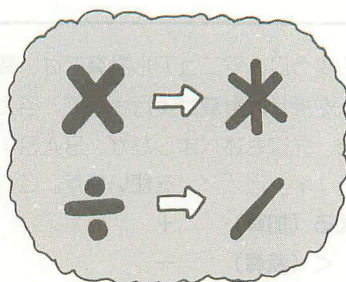
実行モードでは次の図に示しますキー及び機能が働きます。これはプログラムモードでも同じです。



通常の数学では四則計算の記号として「＋，－，×，÷」を使いますが、ＢＡＳＩＣでは「×」や「÷」は使わず、代りに「＊」と「／」を使います。

本機では ×＊ キーを押せば "＊" が入力され、÷／ キーを押せば "／" が入力されます。また、計算結果は ＝ でなく ENTER キーで求めます。

この点に注意してください。



〈 例 〉2+3×4=	
〈操作〉2 [+] 3 [＊] 4	2+3＊4_
[ENTER]	14.
〈 例 〉5×(−6)+7=	
〈操作〉5 [＊] [−] 6 [+] 7	5＊−6+7_
[ENTER]	−23.

●負符号は [−] キーで入力します。この例のように、計算の記号や"("の直後にある"−"記号は減算命令ではなく、負符号と判断されます。

〈 例 〉(6+5)÷(4−1)=	
〈操作〉[(] 6 [+] 5 [)] [/]	
[(] 4 [−] 1 [)]	(6+5)/(4−1)_
[ENTER]	3.666666667
〈 例 〉$-5\times10^{3}\div(4\times10^{-3})=$	
〈操作〉[C·CE] [−] 5 [EXP] 3 [/] 4 [EXP] [−] 3	−5E3/4E−3_
[ENTER]	−1250000.

●指数部を表わすEシンボルは [E] キーでも入力できますが、混乱を避けるため、指数部記号は [EXP] キーで入力するようにしてください。

〈 例 〉520000×43200=	
〈操作〉520000 [＊] 43200 [ENTER]	2.2464E 10 (2.2464×10^{10})

このように、マニュアル計算では、数式通りにキーを押して入力し、最後に [ENTER] キーを押せば計算が実行されて、答えが表示されます。

なお、先にも述べましたが、BASICでは計算の記号として「×」や「÷」は使わず、「＊」と「/」を使います。これらを次にまとめます。

加える（加算）	+
引 く（減算）	−
かける（乗算）	＊（アスタリスク）
割 る（除算）	/（スラッシュ）
符 号	−または+
指 数	E（EXP）
計算の実行	[ENTER]

3．キー操作の訂正

数式などを入れているとき、よく違った数字や命令を入れたり、また抜かしてしまったりするものです。
このようなときは、あわてずに次の方法で訂正しましょう。

(1) 誤りが多いときは、

C·CE キーを押して、今まで入れた内容をすべて消し、あらためて最初から入れ直しましょう。

(2) まちがってほかのキーを押したときは、

◄ キーを押してカーソル※を移動させ、まちがっているキーのシンボルのうえで点滅させます。そして正しいキーを押します。

〈 例 〉2+3＊4を2+3/4と入力した場合 〈操作〉2 ＋ 3 ／ 4	2+3／4_ └まちがっている
	2+3▒4 └ここでカーソルを点滅させます。
	2+3＊▒4 └そして正しいキーを押します。
	2+3＊4_

※カーソル
キー操作をおこなうとき ▬ を表示したり、キー操作の訂正のときに ▓ マークが点滅したりしますが、これらのマークはカーソルと呼ばれるもので、次に操作したキーがこの部分に入ることを示しています。

注）ＢＡＳＩＣモードでは ENTER キーの直前にくる ） を省略することはできません。電卓モード（ＣＡＬモード）では ＝ キーの直前にくる ） は省略することができます。

⑶ まちがって余分なキーを押したときは、

[◄] キーでカーソルを移動させ、余分なキーのシンボルのうえで点滅させます。そして [SHIFT] [DEL ◄] と押します。

（DEL：DELETE（デリート）……削除を意味します。）

〈 例 〉2+3＊4を2+43＊4と入力した場合

〈操作〉	表示
2 [+] 43 [＊] 4	2+43＊4_ └余分なキー
	2+43＊4 └ここでカーソルを点滅させます。
	2+3＊4 4が削除されます。
[ENTER]	14.

⑷ キーの操作が抜けたときは、

[◄] キーでカーソルを移動させ、抜けたキー操作部分の次のキーシンボルのうえで点滅させます。そして [SHIFT] [INS ►] と押すとカーソルの部分に１文字分の空欄ができますので、抜けていたキーを押せば、その位置に入力されます。

（INS：INSERT（インサート）……挿入を意味します。）

〈 例 〉2+3＊4を2+34と入力した場合

〈操作〉	表示
2 [+] 34	2+34_ └＊が抜けています。
	2+34 └ここでカーソルを点滅させます。
	2+3 4 └空欄をつくります。
[＊]	2+3＊4 抜けていたキーを入れます。
[ENTER]	14.

4. プレイバック機能

ENTER キーで計算を実行したのち、◄ あるいは ► キーを押すと実行した式が呼びもどされます。これは実行した式の確認や実行した式の一部を変更して再度実行したいときに便利です。
特に ENTER で計算を実行した結果、エラーになったときにこのプレイバックを使いますと、エラーを起こした場所をカーソルで示していますので、エラーの原因がわかりやすく大変便利です。(>記号(プロンプト記号)が表示されたときには、すでに内容が消されていますのでプレイバックできません。)

〈例〉36/1+2) ENTER と押してエラーになったときのプレイバック

36 / 1 + 2) ENTER	ERROR 1 エラー表示
◄	36/1+2) └カーソル表示

● このあと、訂正をおこないます。この例の場合は36/(1+2)が正しい式なので、"(" を挿入します。

〈例〉36/(1+2) ENTER と押したときのプレイバック

〈操作〉36 / (1 + 2) ENTER	12.
►	36/(1+2) └カーソルが先頭に表示されます。
ENTER	12.
◄	36/(1+2)_ カーソルが最後に表示されます。



5. 連続計算機能

[ENTER] キーを押して計算をおこなったあと、その結果に続けて計算をおこなうことができます。

〈例〉3＋4＝の結果を求め、それを5倍します。

〈操作〉3 [+] 4 [ENTER]

7.

[*] 5

7. *5_

（先の答に続いて計算命令が入力されます。）

[ENTER]

35.

本機は、数字のゼロ(0)をØと表示します。
これは、アルファベットのオー(O)と区別するためで、本書においても、オーとゼロの区別がつきにくい場合はゼロをØと表わしています。

6. ラストアンサー機能

SHIFT CA C・CE と押したあと、↓ キーまたは ↑ キーを押してみてください。
表示部に何かの数字が表示されましたでしょう。
35．が表示されましたか？ それとも別の数が表示されましたか？
じつは、この数値はすぐ前におこなった計算の結果なのです。

本機は、マニュアル計算で ENTER キーを押して得られた結果、あるいは76ページで説明のダイレクトアンサー機能によって得られた結果など（これらをラストアンサーと呼びます）を覚えておく機能があります。（この機能をラストアンサー機能と呼びます。）
そして ↓ キーまたは ↑ キーでラストアンサーを呼び出すことができます。
先に説明しました連続計算機能では、直前におこなわれた計算の結果を式の先頭に入れて計算することができましたが、ラストアンサー機能により、直前におこなった計算の結果を式のどの位置にでも入れることができるようになりました。

〈例〉50÷8＝の結果6.25を使って
12×5÷6.25＋24×3÷6.25＝ を計算します。

操作	表示
〈操作〉50 ／ 8 ENTER	6.25（ラストアンサー）
12 ＊ 5 ／ ↑	12＊5／6.25_（ラストアンサー呼び出し）
＋ 24 ＊ 3 ／ ↓	／6.25＋24＊3／6.25_（ラストアンサー呼び出し）
ENTER	21.12（計算を実行したことによりラストアンサーは入れ替わっています。）
C・CE ↓	21.12_

この例のように、ラストアンサーは何回でも呼び出せますが、次に ENTER キーを押して計算を実行すれば書き替えられます。
なお、ラストアンサーは C・CE や SHIFT CA C・CE の操作などでは消去されません。

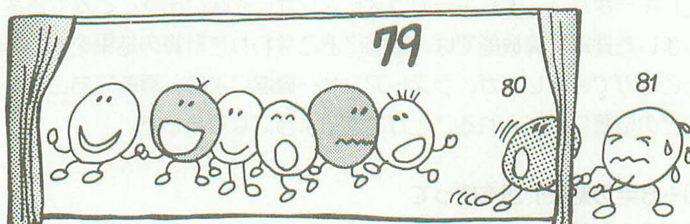
- ラストアンサーは、実行モード（RUNモード）以外のモードになっているときや、プログラムが一時停止している状態のとき、また、トレースモードでは呼び出すことができません。
- ラストアンサーは、プログラムの実行により、計算がおこなわれたときや、表示命令により数値が表示されたときにも書き替えられます。

7. 数式の長さ

これまで、1つの数式の長さなど気にせずに計算してきましたが、無限に長くできるわけではなく、やはり制限があります。

本機では、1回の計算で入力することができる数式の長さ（ ENTER キーを押す前までの長さ）は79キー操作までです。

80キー操作目を入れますと、そのシンボルのうえでカーソル“ ▓ ”が点滅をはじめ、80キー操作目の内容は無効になります。



＝キーやBASIC命令などの記載方法について＝

これ以降の説明では、関数やBASIC命令、計算命令などを表わすとき、特に必要な場合以外は、キーワクなどは省き、計算機で表示されるときと同じ形で表わします。

また、 SHIFT キーなど、ほかのキーに続いて押したとき入力される機能も、特に必要な場合以外は、 SHIFT キーなどの記載を省略します。

例　SHIFT tan⁻¹ (5 / 8) ENTER →ATN (5／8) ENTER と記載

8. 計算結果の表示方法

数値を表わす方法として、通常用いられるのは、たとえば

120000　　や　　0.00123

のような方法ですが、大きな数値や小さな数値は、たとえば

1.2×10^{5}　　や　　1.23×10^{-3}

のような形で表わすことがあります。

数値をこのような形で表わす方法を**浮動小数点方式（指数方式）**と呼びます。

そして、1.2×10^{5}の“1.2”や1.23×10^{-3}の“1.23”を**仮数部**と呼び、“10^{5}”や“10^{-3}”を**指数部**と呼びます。

なお、120000や0.00123のように表わす方法は**固定小数点方式**と呼びます。

本機では、数値を浮動小数点方式で表示する場合は、指数部を示す記号として　**E**　を入れて2.5E12や1.23E−Ø8のように表示します。

1.23E−Ø8

仮数部　　指数部（10^{-8}）

なお、キー入力において、指数部に3桁以上の数値を入れた場合は、後2桁の数字が有効になります。

〈例〉5 EXP 1230 ENTER →5. E　3Ø

エラー表示

マニュアル計算でエラーになりますと

ERROR　1　や　ERROR　2

のようにエラー番号が表示されます。

エラーは C・CE キーで解除することができますが、 ◄ や ► キーでも解除でき、このキーで解除すればエラーが発生した場所が呼び出されます。

（プレイバック機能の説明を参照）

なお、エラー番号の意味は240ページにまとめて記載していますので参照してください。

9. メモリーの使いかた

BASICモードでは、たくさんのメモリーを使うことができますが、くわしい説明は108ページでおこなうことにして、ここではA~Z（A$~Z$）のメモリーを用いて、基本的な説明をおこないます。

それではまず、メモリーAに5を入れてみましょう。次のように操作してください。

操作	表示
A=5 ENTER	5.

メモリーAに5が入ったかどうか確認します。C·CE キーで表示を消去してから次の操作をしてください。

操作	表示
A ENTER	5.

次にメモリーAの内容に3を加えて、その答をメモリーAに入れましょう。

操作	表示
A=A+3 ENTER	8.

次にメモリーBに4×6の答を入れ、Bの内容からAの内容を引いて、その答をメモリーCに入れます。

操作	表示
B=4*6 ENTER	24.
C=B−A ENTER	16.

終りましたら、それぞれのメモリーの内容を確認してみましょう。

操作	表示
A ENTER	8.
B ENTER	24.
C ENTER	16.

こんどはメモリーに文字や記号を入れてみましょう。
文字や記号を入れる場合は、AやBなどのメモリーの名前に$（ドルマーク）を付けて用います。そして、文字や記号は前後を"（クォーテーションマーク：引用符）でくくって示します。
それではメモリーZ$に"CODAMA"を入れ、Y$に"10:26AM"を入れてみましょう。

操作	表示
Z$="CODAMA" ENTER	CODAMA
Y$="10:26AM" ENTER	1Ø:26AM

● "　　" で囲まれた中にある数字は、文字として扱われます。

次にX＄にZ＄の内容を入れます。

X＄=Z＄ ENTER　　　CODAMA

それでは、それぞれのメモリーの内容を確認してみましょう。

Z＄ ENTER　　　CODAMA

Y＄ ENTER　　　10:26AM

X＄ ENTER　　　CODAMA

このように、メモリーに数値や文字を入れたり、呼び出したりすることができます。
最後にX＄とY＄の内容を加えてみましょう。

X＄+Y＄ ENTER　　　CODAMA10:26AM

このように、文字が入っているメモリーどうしで足し算をしますと、前のメモリーの内容に後のメモリーの内容をくっつけます。
ただし、足し算以外の計算はできません。
また、数値が入っているメモリーと文字が入っているメモリーとの間で計算をおこなうことはできません。

ご　注　意

通常“　”（引用符）で囲まれた内容は文字や記号として扱われますが、本機の各キーに定義されています関数命令やBASIC命令を呼び出して“　”内に入れた場合、計算機は関数命令やBASIC命令が誤って文字として指定されたものと見なします。したがって ENTER キーを押したときなどに ERROR 1 を表示します。

例　V SHIFT R($) = SHIFT W(") sin A SHIFT W(")

V$="SIN A"_

ENTER　　　ERROR 1

もし、文字として“　”内に関数やBASIC命令を入れる場合は、アルファベットキーで1文字ずつ入れてください。

例　V SHIFT R($) = SHIFT W(") S I N SPC A SHIFT W(")

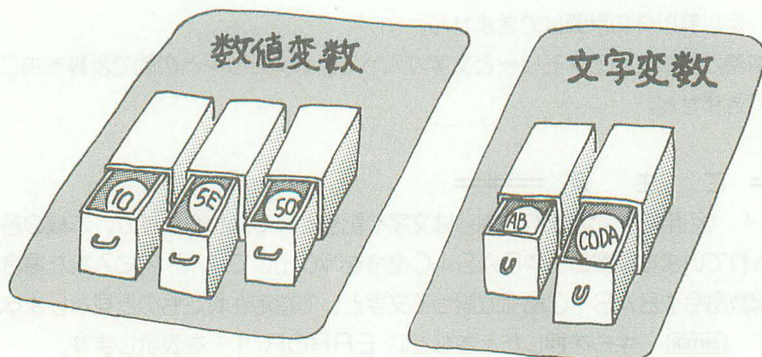
V$="SIN A"_

ENTER　　　SIN A

10. データメモリーと変数

これまで、メモリーに数値や文字などのデータを入れたり、呼び出したりする方法を説明しましたが、このようなデータを扱うメモリーを**データメモリー**と呼びます。
さて今、たとえばデータメモリーAを使ってA＋3という式を作ったとします。
この場合、Aの内容を5、10、17などのように変えても同じ式で計算できるわけです。
このように、式の中などでデータメモリーの内容を変化させることができることから、データメモリーのことを**変数**と呼びます。
変数は、次のように分類されます。

変数（データメモリー）
- 数値変数（数値メモリー）：10、500、5Ｅ12など、数値を入れます。
- 文字変数（文字メモリー）：ＡＢＣ、ＣＯＤＡＭＡなど、文字列を入れます。



10－1）変数の名前

変数の名前はＡ～Ｚだけでなく、ＡＡ、ＢＣ、Ａ５などのように、文字・数字を２文字使用することができます。
この場合は、最初の文字を必らず英文字にしてください。
なお、変数の名前にＮＡＭＡＥやＫＯＴＡＥのように３文字以上を使用した場合でも、計算機は最初の２文字を変数名と判断します。
したがって、たとえば変数名にＫＯＴＡＥ１とＫＯＴＡＥ２を使用しますと、計算機は区別できず、ＫＯという同じ変数と判断してしまいます。
また、本機が持っている関数命令やＢＡＳＩＣ命令は変数名として使用できません。

変数へのデータの入れかたと、その大きさ

① 数値変数に数値を入れる方法

変数に数値を入れる場合は次の形で指定します。

数値変数＝式　（式とは、数値、数式、数値変数を指します。）

〈例〉 A＝3 ENTER　…変数Aに3が入ります。
B＝A＋2 ENTER　…変数Aの内容に2を加えた結果が変数Bに入ります。
AA＝5＊B ENTER　…5と変数Bの内容との乗算結果が変数AAに入ります。

そして数値変数には仮数部10桁、指数部−99〜99までの数値を入れることができます。

② 文字変数に文字を入れる方法

変数に文字を入れる場合は次の形で指定します。

文字変数＝"文字"　（変数名の後に＄(ドルマーク)をつけることにより
文字変数＝文字変数　文字変数を表わします。）

〈例〉 A＄＝"SHARP" ENTER　…A＄にSHARPが入ります。
AB＄＝"COMPUTER" ENTER　…AB＄にCOMPUTERが入ります。

そして、文字変数には次の文字数を入れることができます。

- 1文字の変数…最大7文字まで入れることができます。
- 2文字の変数…最大16文字まで入れることができます。

注）変数のうち、数値変数A〜Zと文字変数A＄〜Z＄は同じメモリーが使用されます。たとえば、数値変数Aと文字変数A＄は同じ場所を使用します。したがって変数Aと変数A＄が同時に存在することはありません。

〈例〉 A＝123 ENTER　→変数Aに123を入れます。
A＄＝"ABC" ENTER　→変数Aを消去し、A＄にABCを入れます。
A＄ ENTER　→ABC表示
A ENTER　→エラー

（計算機は文字変数が数値変数の形で指定されたものとしてエラー表示します（ERROR　9））

なお、2文字変数の場合、たとえばAAとAA＄は同時に用いることができます。

11. 関数計算

本機は、ＢＡＳＩＣモードで使える基本関数が多くあります。
これらの基本関数を次に示しますが、ＢＡＳＩＣでは基本的に、数式を入力して ENTER キーで実行します。したがって、関数命令が表示されますが、通常の数学などとは表現のしかたが異っていますので、ご注意ください。

項目	一般の表現	本機の表現	入力方法	備考
三角関数	sin cos tan	SIN COS TAN	sin cos tan	
逆三角関数	$\sin^{-1}$ $\cos^{-1}$ $\tan^{-1}$	ASN ACS ATN	SHIFT $\sin^{-1}$ SHIFT $\cos^{-1}$ SHIFT $\tan^{-1}$	
双曲線関数	sinh cosh tanh	HSN HCS HTN	hyp sin hyp cos hyp tan	
逆双曲線関数	$\sinh^{-1}$ $\cosh^{-1}$ $\tanh^{-1}$	AHS AHC AHT	SHIFT archyp sin SHIFT archyp cos SHIFT archyp tan	
対数関数	ln log	LN LOG	ln log	$\log_e x$ （ｅを底とする対数） $\log_{10} x$ （10を底とする対数）
指数関数	e^x 10^x	EXP TEN	SHIFT e^x SHIFT 10^x	e ≒2.718281828
逆数	$\frac{1}{x}$	RCP	$1/x$	
2乗	x^2	SQU	x^2	
平方根	$\sqrt{\ }$	$\sqrt{\ }$ あるいは SQR	$\sqrt{\ }$	
立方根	$\sqrt[3]{\ }$	CUR	SHIFT $\sqrt[3]{\ }$	
階乗	n！	FACT	SHIFT n！	
円周率	π	π あるいは PI	SHIFT π	π ≒3.141592654

項目	一般の表現	本機の表現	入力方法	備考
60進数(度分秒)→10進数(度)変換		DEG	[→DEG]	
10進数(度)→60進数(度分秒)変換		DMS	[SHIFT] [→D.MS]	
べき乗	y^x	^	[SHIFT] [^] または [y^x]	y ^ x において y^x
べき乗根	$\sqrt[x]{y}$	ROT	[SHIFT] [$\sqrt[x]{y}$]	y ROT x において $\sqrt[x]{y}$
10進→16進変換		HEX	[→HEX]	74ページ参照
16進→10進変換		DEC I あるいは&	[SHIFT] [→DEC] [SHIFT] [&]	74ページ参照
直交座標→極座標変換		POL	[SHIFT] [→$r\theta$]	POL(x, y)においてr, θ を求めます。
極座標→直交座標変換		REC	[SHIFT] [→xy]	REC(r, θ)においてx, y を求めます。
整数化 (インテジャー)		INT	[I] [N] [T]	INT(x)において、x以下でかつもっとも大きい整数を求めます。
絶対値 (アブソリュート)	$\|x\|$	ABS	[A] [B] [S]	ABS(x)において、xの絶対値を求めます。
符号関数 (シグナム)		SGN	[S] [G] [N]	SGN(x)において、$x>0$のとき1、$x<0$のとき−1、$x=0$のとき0が得られます。

これらの関数のうち、INT、ABS、SGNはアルファベットキーで入力しますが、そのほかの関数も、アルファベットで表現されるものはアルファベットキーで入力することができます。
たとえばsin30 を入力するときは [sin] 30と操作することも [S] [I] [N] 30 と操作することもできます。

なお、三角関数、逆三角関数および座標変換の計算では、取り扱う角度の単位を正しく指定しておく必要があります。
マニュアル計算では、電卓モードのときと同じように [SHIFT] [DRG] の操作で指定することができますが、次の命令で指定することもできます。

角度単位	命　令	表示シンボル	備　考
度	DEGREE	D E G	直角を90で表わす単位(°)
ラディアン	RADIAN	R A D	直角を$\frac{\pi}{2}$で表わす単位(rad)
グラード	GRAD	GRAD	直角を100で表わす単位(^{g})

プログラム中では、この命令で指定しますので、練習のため、次の計算例ではこの命令により角度単位を指定してください。

注）計算例で特に記載しませんが、関数計算の式を入力するとき、C·CE キーで以前の計算結果などを消去してから、新しく式を入力するようにしてください。
計算結果などが表示されているときに関数キーを押しますと、76ページで説明しますダイレクトアンサー機能が働いて、計算例通りの結果が得られない場合があります。

〈例〉 sin30°＝
〈操作〉 DEGREE ENTER　（角度単位を"度"に指定）
SIN30 ENTER　　Ø.5

〈例〉 tan$\frac{\pi}{4}$＝
〈操作〉 RADIAN ENTER　（角度単位を"ラディアン"に指定）
TAN(π／4) ENTER　　1.

〈例〉 $\cos^{-1}(-0.5)$
〈操作〉 DEGREE ENTER　（角度単位を"度"に指定）
ACS−0.5 ENTER　　12Ø.
(120°)

〈例〉 log5＋ln5＝
〈操作〉 LOG5＋LN5 ENTER　　2.3Ø84Ø7917

〈例〉 e^{2+3}＝
〈操作〉 EXP(2＋3) ENTER　　148.4131591

〈例〉 $\sqrt[3]{4^3+5^3}$ ＝
〈操作〉 CUR(4^3＋5^3) ENTER　　5.738793548

〈 例 〉60進数の30度30分は10進数の何度？

〈操作〉DEG30.30 ENTER

30.5

(30.5度)

〈 例 〉10進数の30.755度は60進数の何度何分何秒？

〈操作〉DMS30.755 ENTER

30.4518

(30度45分18秒)

〈 例 〉直交座標→極座標変換
直交座標で点Pの座標が(3，8)の
場合の極座標（r，θ）を求めます。

〈操作〉DEGREE ENTER　（角度単位を"度"に指定）

POL(3,8) ENTER　(r) 8.544003745

(r≒8.5)

Z ENTER　(θ) 69.44395478

(θ≒69°)

●θ の値は変数Zに収められています。また、r の値は変数Yに収められています。

〈 例 〉極座標→直交座標変換
極座標で点Pの座標が$(12, \frac{4}{5}\pi)$ の
場合の直交座標(x，y)を求めます。

〈操作〉RADIAN ENTER　（角度単位を"ラディアン"に指定）

REC(12,(4/5＊π)) ENTER　(x) −9.708203933

(x≒−9.7)

Z ENTER　(y) 7.053423028

(y≒7.1)

●y の値は変数Zに収められています。また、x の値は変数Yに収められています。

注）座標変換では、計算結果を変数ZとYに入れますので、それまでZ、Y（あるいはZ$、Y$）に入っていた内容は消されます。

〈 例 〉10進→16進変換

1250を16進数に変換します。

〈操作〉HEX 1250 ENTER

4E2.

SHIFT CA

BASICモードでは10進→16進変換（HEX命令）を使用した連続計算はできません。

10進→16進変換をおこなった後は必らず SHIFT CA と押してから、次の計算をおこなってください。

10+HEX20 ENTER

30.

● HEX命令を式の中に入れて実行した場合、1度16進数に変換された値は、再度10進数にもどされ、10進数で計算がおこなわれます。

HEX1.25 ENTER

ERROR 1

● 16進数に変換しようとする数値は整数でなければなりません。

小数部を含んでいる場合はエラーになります。

〈 例 〉16進→10進変換

16進数CF8を10進数に変換します。

〈操作〉DECI CF8 ENTER または

& CF8 ENTER

3320.

● 16進数字（0〜9、A〜F）のA〜Fはアルファベットキーで入力してください。（電卓モードとは異なります。）

DECI 1.25 ENTER

ERROR 1

● 16進数は整数のみ扱います。

したがって小数部を含む数値を変換しようとするとエラーになります。

＝参　考＝

論理演算子(=、>、<、>=、<=、<>)を用いて構成された式（論理式）は、それぞれ次の値をとります。

x、y は数値を示します。

=※	$x=y$ のとき：1 $x \neq y$ のとき：0	>=	$x \geqq y$ のとき：1 $x < y$ のとき：0
>	$x > y$ のとき：1 $x \leqq y$ のとき：0	<=	$x \leqq y$ のとき：1 $x > y$ のとき：0
<	$x < y$ のとき：1 $x \geqq y$ のとき：0	<>	$x \neq y$ のとき：1 $x = y$ のとき：0 （<>は≠と同意味）

※＝を用いる論理式を例えばA＝数値、B＝式のような形にしますと、計算機は論理式と判断せず、変数への入力(代入文)と判断しますので、このような場合は数値＝A、式＝Bのような形にする、あるいはカッコでくくる、などしてください。ただし、ＩＦ文における条件式の場合はこの限りではありません。

12. ダイレクトアンサー機能

これまでのマニュアル計算では「式の入力 ENTER 」の操作で計算を実行しましたが、この計算機では、BASICモードでも電卓モードと同じく、置数のあとや計算のあと、数値が表示されているときに関数キーを押せば、直接関数計算を実行することができます。

ただし、この機能は、べき乗やべき乗根、座標変換のように2つの数値が必要な関数(2項関数)では働きません。

〈 例 〉 sin30° と8! を計算します。

〈操作〉 DEGREE ENTER

C·CE 30	3Ø_
sin	Ø.5
C·CE 8	8_
SHIFT n!	4Ø32Ø.

〈 例 〉 $\tan^{-1}\frac{5}{12}$ において $\frac{5}{12}$ の結果を確認したあと、$\tan^{-1}$ を求めます。

〈操作〉 DEGREE ENTER

5/12 ENTER	4.166666667E-Ø1
SHIFT tan⁻¹	22.61986495

このダイレクトアンサー機能は、式の入力途中などでは働きません。

〈 例 〉 C·CE 5*4 →5*4_

log →5*4LOG _

●ダイレクト計算をおこなった場合は ◄ キー、► キーを押してもプレイバック機能は働かず、カーソルのみが表示されます。

13. 計算の優先順位

BASICモードではカッコ、関数も含めて数式通りのキー操作で計算をおこなうことができます。計算の優先順位の判断や途中結果の処理はすべて計算機が自動的に処理してくれます。計算の優先順位は次の通りです。

1．π や変数の呼び出し
2．関数（SIN、COSなど）
3．べき乗（^）、べき乗根（ROT）
4．符号（+、−）
5．乗除算（*、／）
6．加減算（+、−）
7．大小比較（>、>=、<、<=、<>、=）
8．AND、OR

注）●カッコが使用されている場合はカッコ内の計算が最優先されます。

●複合関数（sin $\cos^{-1}$ 0.6など）は右から左の順で計算されます。

ただし、HEX、DECI（あるいは&）は関数の中でも優先順位が高くなり、これらが複合した場合はHEX、DECI（あるいは&）の計算から実行されます。

例）DECI　SQU 5 [ENTER] → ERROR 1

このような場合はカッコでくくってください。

DECI （SQU 5）[ENTER] → 37.

●べき乗、べき乗根の連算（3^{4^2} すなわち3^4^2など）は右から左の順で計算されます。

●上記3.と4.では後に出てきたほうが優先順位が高くなります。

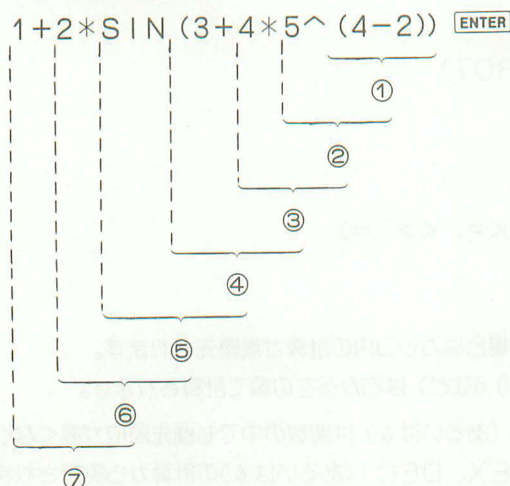
例）−2^4 は −（2^4）となります。

3^−2 は 3^{-2} となります。

＝参　考＝

本機では数式通りの計算をおこないますが、数式通りの計算をするためには、すぐに処理できない命令やデータ(数値)を一時記憶しておく場所が必要になります。
この一時記憶の場所をバッファといい、本機ではファンクション用バッファを16段、データ用バッファを８段持っています。

〈例〉１＋２＊ＳＩＮ（３＋４＊５^（４−２））　ENTER　と押したときの計算順序と、
バッファの使用状況



データバッファ							ファンクションバッファ									
1段	2段	3段	4段	5段	6段	………	1段	2段	3段	4段	5段	6段	7段	8段	9段	………
4	5	4	3	2	1		−	(	^	＊	＋	(	SIN	＊	＋	

例では①→⑦の順で計算が実行されますが、①の計算をおこなうために、②以降の計算を上図のように保存します。
そして、１段目から、計算用のメモリーに呼び出して計算を進めます。
上例のようにカッコもファンクションバッファに記憶されます。
なお、カッコは最高15段までなら、ファンクションバッファを越えない範囲で深みを持たせることができます。（電卓モードの場合とは少し異なります。）

第4章

プログラム

■もくじ■

（ページ）
1．プログラム・その1……………… 81
1−1）プログラムの形…………… 81
1−2）プログラムの構成………… 83
1−3）プログラムの入れかた…… 84
1−4）プログラムの修正・編集… 86
1−5）プログラムの実行………… 90
1−6）エラー表示と処理方法…… 91
1−7）プログラム作成の手順…… 92
2．プログラム・その2……………… 95
2−1）入力命令（INPUT）…… 95
2−2）出力命令（PRINT）…… 98
2−3）代入文（LET）……………100
2−4）終了文（END）……………100
2−5）ジャンプ（GOTO）………100
2−6）判断（IF…THEN〜）…103
2−7）くり返し
（FOR…TO…
STEP, NEXT）……104
3．変数………………………………108
3−1）変数の構成……………………108
3−2）固定変数………………………108
3−3）単純変数………………………109
3−4）配列変数………………………110
3−5）メモリーの構成と変数……117
3−6）変数をクリアするには
（CLEAR）……………118
3−7）変数A（　）について………119
4．定義付けプログラム………………121
5．サブルーチン（GOSUB,
RETURN）……………………122

(ページ)
6．表示に関するBASIC…………125
6−1）フォーマット指定
(USING)…………………125
6−2）表示時間の指定
(WAIT) ……………………128
6−3）ポーズ(PAUSE)………128
7．そのほかのBASIC…………130
7−1）READとDATA、
RESTORE……………130
7−2）ON GOTO、
ON GOSUB…………132
7−3）REM(リマーク)…………133
7−4）STOP(ストップ)………133
7−5）CONT
(コンティニュー)……………133
7−6）AREAD
(オートリード)………………134
8．特別な機能……………………………135
8−1）乱数関数
(RND, RANDOM)…135
8−2）INKEY$
(インキードル)…………136
8−3）音発生機能(BEEP)……137
8−4）MEM(メム)………………138
8−5）論理積、論理和、否定
(AND,OR,NOT)……139
8−6）条件式の結合
(AND, OR)……………141
9．文字の編集……………………………143
9−1）文字列の分断
(LEFT$,RIGHT$,
MID$) …………………143
(ページ)
9−2）文字列の結合………………144
9−3）文字数を数える方法
(LEN)…………………………145
9−4）数値と文字の変換
(VAL, STR$)………146
9−5）アスキーコードの変換
(ASC, CHR$)………147
9−6）文字列の比較………………148
10．パスワード(PASS)……………149
11．デバッグ………………………………150
12．プリンタに関する命令………………152
12−1)プリント命令
(LPRINT)……………152
12−2)PRINT→LPRINT
指定……………………………153
12−3)マニュアル計算での
プリント……………………154
12−4)リストのしかた
(LLIST)……………………155
13．テープレコーダーに関する命令…156
13−1)プログラムの記録命令
(CSAVE)……………………156
13−2)プログラムの転送命令
(CLOAD)……………………158
13−3)照合命令(CLOAD?)…159
13−4)データの記録命令
(PRINT#)…………………160
13−5)データの転送命令
(INPUT#)…………………162
14．BASICモードでの
キーの機能……………………165

1. プログラム・その1

プログラムとは、計算機に計算などをおこなうための手順を指令する命令書のようなものです。
この指令を計算機が理解できるように書き表わしたり、計算機に覚えさせたりすることをプログラミングとか、プログラムを組むなどと言います。
ここでは、プログラムの構成、入れかた、編集方法など、プログラミングの基本について説明します。

1－1）プログラムの形

ここでは、プログラムの基本的な形について説明します。まず、次のプログラムを見てください。

〈プログラム1〉

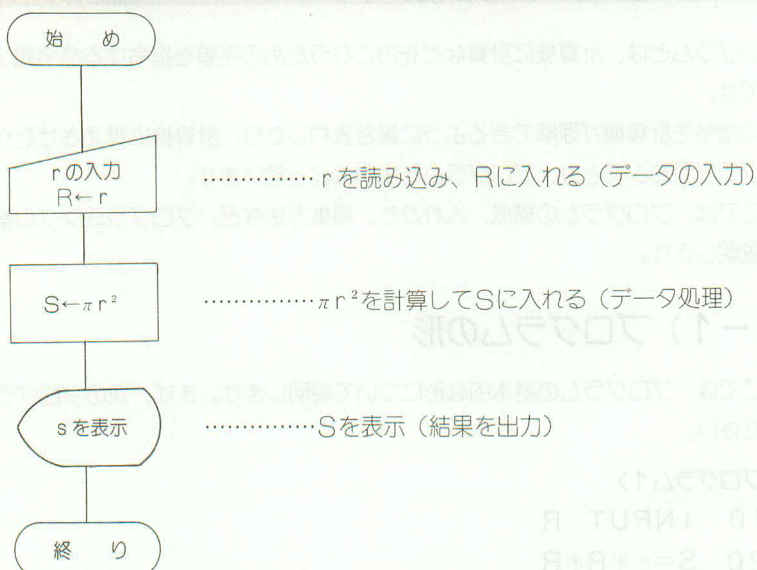
```
10  INPUT R
20  S=π*R*R
30  PRINT S
40  END
```

このプログラムは、半径を入力して円の面積を求めるプログラムです。
BASICをまったく知らない人でもこのプログラムが計算機に何を伝えているのか想像がつくと思います。これは次のような指示をするプログラムなのです。

```
10  INPUT R    ←データを読み込み、Rに入れなさい。
20  S=π*R*R    ←πにRの値を2回掛けなさい。そして、その結果をSに
                 入れなさい。
30  PRINT S    ←Sの値を表示しなさい。
40  END        ←これでプログラムは終りです。
```

これをフローチャート（流れ図）で表わしますと、次のようになります。

（フローチャート）



フローチャート（流れ図）を見ていただければ、おわかりのように、プログラムの大部分は

① データを読み込み、
② データを処理して、
③ 結果を出力する。

という形になっています。

どんなに長いプログラムであっても、基本的には上記①～③のくり返し、あるいは①、②、③それぞれを何回かくり返す、という形になっています。
したがって、プログラムはこの形が基本になります。

1－2）プログラムの構成

さて、ここでプログラムの構成について述べます。
先ほどのプログラムを見てみましょう。

```
ライン(行)
  →10 INPUT R··············10ライン(10行)
  →20 S=π＊R＊R·············20ライン(20行)
  →30 PRINT S··············30ライン(30行)
  →40 END    ··············40ライン(40行)
    ↑   ステートメント(文)
ラインナンバー(行番号)
```

このようにBASICのプログラムは、いくつかのライン、ラインナンバー、ステートメントで構成されています。
これらについて、次にくわしく述べます。

(1) ライン（行）

上のプログラムで、横一行の並びをライン（行）と呼びます。
1つのラインは、1つのラインナンバーと、いくつかのステートメントからなります。
そして、ラインの終りは ENTER キーで指定されます。

```
______  ________________ ENTER
ラインナンバー  ステートメント
```

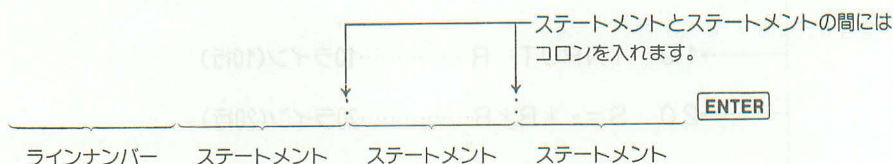
なお、1つのラインの長さは ENTER キーも含めて、最大80キー命令（あるいは最大80バイト）までの長さにすることができます。

(2) ラインナンバー（行番号）

ラインナンバーには1～65279までの整数を使用することができます。
プログラムの実行は若いラインナンバーから順次実行されますので、実行させたい順番に若いラインナンバーをつけます。ただし、あとからプログラムの途中にラインを追加したい場合も出てきますので、そういったことを配慮して、10、20、30……のように10番おきにラインナンバーをつけるのが最も一般的です。

(3) ステートメント（文）

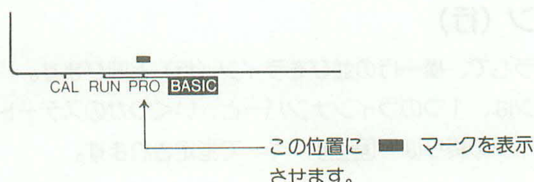
1つの意味を持った処理式や命令をステートメントといいます。
1つのラインは1つ以上のステートメントからなり、2つ以上のステートメント（マルチステートメント）になる場合は、ステートメントとステートメントの間に：(コロン）を入れて区別します。



1－3）プログラムの入れかた

81ページのプログラムを計算機に書き込み、覚えさせてみましょう。
プログラムを計算機に書き込むときは、次の手順でおこないます。

① まず、電源スイッチをON位置にして電源を入れ、BASIC キーでプログラムモード（PROモード）にしてください。



② 次にNEW ENTER と押してください。
これで、先に記憶されていた（かもしれない）プログラムやデータがすべて消されます。

③ 続いてプログラムを1ラインずつ入れてください。
たとえば、10ラインを入れるときは次のように押します。

キー操作	表示
1 0 SHIFT A(INPUT) R	10 INPUT R_
ENTER	10: INPUT R

または

キー操作	表示
1 0 I N P U T R	10INPUTR_
ENTER	10: INPUT R

このとき、注意していただきたいことは、1ライン入れ終るごとに [ENTER] キーを押すことです。

[ENTER] キーは、ラインの終りに“このラインは終りました”という意味で必らず必要ですので、忘れないでプログラムを入れてください。

〈書き込み操作例〉

PROモード指定

操作	表示
NEW [ENTER]	〉
10INPUTR [ENTER]	10:INPUT R
20S=π*R*R [ENTER]	20:S=π*R*R
30PRINTS [ENTER]	30:PRINT S
40END [ENTER]	40:END

INPUTは [SHIFT] [A](INPUT) または [I] [N] [P] [U] [T] と押して入力します。
PRINTは [SHIFT] [Z](PRINT) または [P] [R] [I] [N] [T] と押して入力します。
ENDは [SHIFT] [N](END) または [E] [N] [D] と押して入力します。

プログラムを入れていただければ、おわかりと思いますが、各ラインを入れて [ENTER] キーを押しますと、ラインナンバーの後に自動的に：（コロン）が入ります。
また、INPUTやPRINTを1文字ずつ入れたときは、これらの命令の後はつまっていますが、 [ENTER] キーを押すと、自動的に1桁あけられます。(これらの命令がラインの先頭ではなく、途中にある場合は、その前の命令や変数、記号などの間も1桁分あけられる場合があります。)

10:INPUT␣R

このことにより、計算機が最初の数字をラインナンバーとして受け取り、アルファベットを1つの命令（INPUTという命令）として受け取り、覚えたことを示します。
ただし、 [SHIFT] [A](INPUT) と押して入力した命令の後は、入力したときからあけられています。

1−4）プログラムの修正・編集

プログラムを作成する場合、一度で正しくは、なかなか書けないものです。キー操作のミスや勘違いでまちがえることもあります。このような場合の処理方法について次に説明します。

(1) プログラムの確認

プログラムを計算機に書き込んだあと、まちがいがないか確認してください。

確認は [↑] キー及び [↓] キーの操作、ＬＩＳＴ（リスト）命令によりおこなうことができます。

① [↓] キーの働き

このキーを１回押すごとに現在表示しているラインの次のラインを表示します。また、このキーを押し続けますと、順次、次のラインを表示します。

[SHIFT] [CA] などにより計算機がクリアされているときは、[↓] キーで先頭のラインを表示します。

② [↑] キーの働き

このキーを１回押すごとに現在表示しているラインの前のラインを表示します。また、このキーを押し続けますと、順次前のラインを表示します。

[SHIFT] [CA] などにより計算機がクリアされているときは、[↑] キーで最後のラインを表示します。

③ ＬＩＳＴ（リスト）命令の働き

ＬＩＳＴ命令は指定したプログラムラインを呼び出す命令で、プログラムモードで

ＬＩＳＴ[ENTER]

と押せば、最も若いラインを表示します。また、

ＬＩＳＴ＿＿＿＿＿＿＿＿ [ENTER]
　　　　ラインナンバー

の形で実行すれば、指定したラインを表示します。

〈例〉ＬＩＳＴ20 [ENTER]　　　　20：S＝π＊R＊R

もし、指定したナンバーのラインがない場合は、それよりも大きく、かつ一番近いラインが表示されます。

なお、ＬＩＳＴ命令はプログラムモード（ＰＲＯモード）でのみ働きます。

●表示部には最大16桁表示することができますが、１ラインに長いプログラムを書きますと、その一部は表示できなくなります。このような隠れた部分を見るときは [►] キーを押し続けてください。そうすればカーソルが右端まで行き、そののち表示が左に送られて隠れている部分が出てきます。左に隠れた部分を見るときは [◄] キーを押します。

(2) 現在表示されているライン内の修正は

表示されているライン内の訂正や追加、削除は59ページのマニュアル計算での説明と同じように [DEL ◀] または [INS ▶] キーでカーソルを移動させ、[SHIFT] [DEL ◀] や [SHIFT] [INS ▶] を使っておこないます。

〈例〉20 S=π*R*R を
20 π*R*R=S [ENTER] と押した場合の訂正

〈操作〉	表示
20π*R*R=S [ENTER]	20:π*R*R=S
[INS ▶] または [DEL ◀] を操作	20 π*R*R=S └カーソルをここで点滅させます。
[SHIFT] [INS ▶] を操作	20 ▒□π*R*R=S └必要な桁数分をあけます。
[S] [=]	20 S=π*R*R=S └追加します。
[INS ▶] を操作	20 S=π*R*R▒S ↑カーソルをここで点滅させます。
[SHIFT] [DEL ◀] [SHIFT] [DEL ◀]	20 S=π*R*R_ ↑不要な内容を削除します。
[ENTER]	20:S=π*R*R

(3) ラインの追加、変更、削除は

① プログラムを入力するとき、1ライン抜かしてしまったときや、新しいラインを追加したいときは、次の方法で追加してください。

〈例〉プログラム1に、次のようにPRINT　R　を追加したい場合

```
10  INPUT  R
20  S=π*R*R        ←この間に PRINT  R  を追加したい。
30  PRINT  S
40  END
```

20ラインと30ラインの間に追加する場合は、この間のラインナンバー（21～29のいずれか）を使ってプログラムを書きます。たとえば

25PRINTR [ENTER]

と操作します。

そうすればプログラムは次のようになります。（ [↓] キー及び [↑] キーで確認してください。）

```
10  INPUT  R
20  S=π*R*R
25  PRINT  R
30  PRINT  S
40  END
```

この例でおわかりのように、プログラムは若いラインナンバーから書く必要はなく、計算機が自動的に若いラインナンバーから順に並べてかえてくれます。

② 不要になったラインを削除したいときは次の方法でおこなってください。

〈例〉先に追加した25ラインを削除します。

〈操作〉25 [ENTER]

>

[↓] キー及び [↑] キーでプログラムを確認してください。

このように1ラインすべてを削除するときは、**ラインナンバー** [ENTER] と押します。

③ ラインナンバーを変更するときは次の方法でおこなってください。

〈例〉プログラム1の30ライン、40ラインをそれぞれ40ライン、50ラインに書き替えます。

〈操作〉LIST40 ENTER	40:END └40ラインを表示させます。
◄ を操作	40 END └カーソルをここで点滅させます。
5 ENTER	50:END 50ラインが書き込まれます。
LIST30 ENTER	30:PRINT S └30ラインを表示させます。
◄ を操作	30 PRINT S └カーソルをここで点滅させます。
4 ENTER	40:PRINT S 40ラインが書き込まれます。 (以前の40ラインは削除されます。)
30 ENTER (不要になった30ラインを削除します。)	>

●この例のように、すでに書き込まれているラインのラインナンバーを変えて書き込んだ場合、変える前のラインはそのまま残っていますので、不要な場合は削除する必要があります。

1−5）プログラムの実行(RUN)

プログラムを入れ、確認や修正が終りましたらプログラムを実行してみましょう。

プログラムの実行は、実行モード（RUNモード）でおこないますので BASIC キーで指定してください。



この位置に ▬ マークを表示させます。

そして、次の命令で実行を開始します。

RUN ENTER（最も若いラインナンバーより実行を開始）

RUN ＿＿＿＿ ENTER（指定したラインナンバーより実行を開始）

ラインナンバー



それでは、プログラム1（81ページ）を実行してみましょう。

〈例〉①半径が5㎝の円の面積は？

②半径が10㎝の円の面積は？

〈操作〉	RUN ENTER	（実行開始）	? └データの入力待ち（rの値は？）
	5	（データ入力）	5_ （5を入力）
	ENTER	（実行再開）	78.53981634 結果を表示（Sは約78.5㎠）
	ENTER	（実行再開）	> └プログラムの実行終了

RUN	ENTER	(実行開始)	? └データの入力待ち
10		(データ入力)	10_ (10を入力)
ENTER		(実行再開)	314.1592654 結果の表示(Sは約314.2㎠)
ENTER		(実行再開)	〉 └プログラムの実行終了

このように、プログラムの実行を開始するとINPUT命令により、?を表示して実行をいったん停止します。
このとき、データを入力して ENTER キーを押せば、実行が再開され、計算がおこなわれて、その結果がPRINT命令により、表示されます。
このときも、プログラムの実行がいったん停止しますので、ENTER キーで再開させます。
そして、プログラムの実行が終了しますと、計算機は 〉シンボル (プロンプト記号) を表示します。

1−6) エラー表示と処理方法

プログラムを実行したとき、プログラムに誤りがあったり、データが不適当な場合などにエラーが発生することがあります。
エラーが発生しますと、次のようなエラーメッセージが表示されます。

ERROR 9 IN 20

エラー番号┘ └ラインナンバー

このエラーメッセージは、エラーの種類と、エラーが発生したラインを示しています。

もし、エラーが発生した場合は、240ページの表をご覧いただいてエラー内容を確認し、次のようにエラー処理をおこなってください。

① C·CE キーでエラーを解除してください。
(エラーメッセージが表示されているとき ↑ キーを押せば、押している間だけエラーが発生したプログラムラインを表示します。)

② BASIC キーでプログラムモード (PROモード) を指定してください。

③ ↓ キーまたは ↑ キーを押してください。エラーが発生した位置がカーソルで示されています。

④ エラーが発生した原因を探して、プログラムを修正してください。

1－7）プログラム作成の手順

これまで、プログラムの構成や編集、プログラムの入れかたや実行方法などについて説明しました。
ここで、プログラムを作成する場合の手順について述べておきます。

プログラムは、大体次の手順で作っていきます。

① まず、目的を決めます。
たとえば、「三角形の面積を求める」というように何をするか決めます。

② そして、目的（問題）を分析します。
たとえば、三角形の面積を求めるために、どのデータが得られるか、得られたデータから、どの式で計算することができるか、などを分析します。

③ 次に、目的を達成するための手順をフローチャート（流れ図）などに表わします。

④ 手順にしたがって、ＢＡＳＩＣ命令でプログラムを紙に書いていきます。

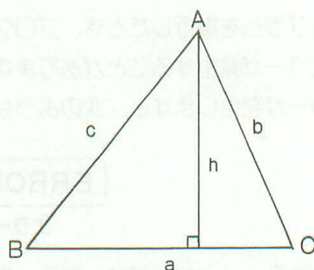
⑤ プログラムを計算機に入れ、実行しながら誤りや不都合な点がないかチェックし、修正などをおこないます。

以上の手順でプログラムを作成します。

それでは「三角形の面積を求める」という目的で、プログラムを作成してみましょう。

① 三角形の面積を求める。

② 右の図において、面積を求める場合どの角や辺の値が得られるかによって、いろいろな式で計算することができます。
たとえばａとｈが得られれば



$$S=\frac{1}{2}ah$$

で計算できます。
したがって、どれがデータとして得られるか、また、得られたデータから、どの式を使って計算できるかなどを分析します。
ここでは、辺ｃ、ａと角Ｂの値が得られるものとして、計算式は、

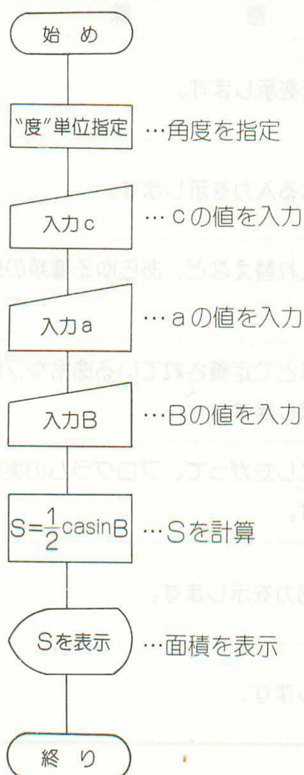
$$S=\frac{1}{2}ca\sin B$$

を用いるものとします。（角度Ｂは “度” の単位とします。）

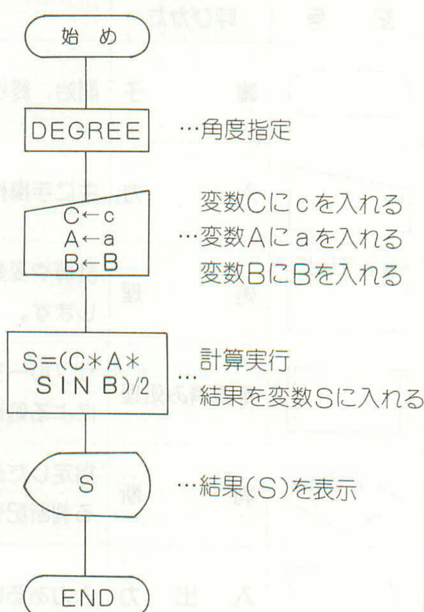
③ 次に計算をおこなう手順を考えます。

この手順はフローチャート（流れ図）で表わします。

（フローチャート1）



（フローチャート2）



●フローチャート2はフローチャート1をよりプログラムに近い形にしたものです。手順が複雑になればなるほど、何段階かにわけて書いたほうが、わかりやすくなります。

④ このようにフローチャートを書けば、あとは手順通りにプログラムを書いていきます。

```
10 DEGREE
20 INPUT C
30 INPUT A
40 INPUT B
50 S=(C*A*SIN B)/2
60 PRINT S
70 END
```

⑤ プログラムを計算機に入れ、誤りがないかチェックし、誤りや不都合な点があれば修正して、プログラムの作成を終了します。

⊙フローチャート（流れ図）記号

フローチャートに用いられる記号で一般的によく使われるものの意味を次に示します。

記　号	呼びかた	意　　味
（端子の記号）	端　　子	開始、終りなどを示します。
（入力の記号）	入　　力	主に手操作による入力を示します。
（処理の記号）	処　　理	計算や変数の入れ替えなど、あらゆる種類の処理を示します。
（定義済み処理の記号）	定義済み処理	サブルーチンなどで定義されている命令やプログラムによる処理を示します。
（判断の記号）	判　　断	指定した条件にしたがって、プログラムの実行を変える判断記号です。
（入出力の記号）	入　出　力	入力あるいは出力を示します。
（表示の記号）	表　　示	表示出力を示します。

2. プログラム・その2

この項では、プログラムの中で特によく使われ、基本になる命令を重点的に説明します。

2−1）入力命令（INPUT）

INPUT命令は、プログラムの実行を一時止め、マニュアル操作により数値や文字の入力をおこなうための命令です。
この命令は次の(1)～(3)の形で用いることができます。

(1) INPUT 変数，変数……

INPUT A	変数Aに数値を入れなさい
INPUT C$	変数C$に文字を入れなさい
INPUT D, E$, F	複数の入力の指定

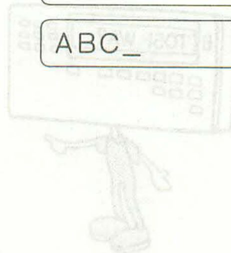
〈例〉
```
10 INPUT A
20 INPUT B, C$
      :
```

●計算機に書き込んで実行してみてください。

実行モード（RUN）

操作	表示
RUN ENTER	?
10	1Ø_
ENTER	?
25	25_
ENTER	?
ABC	ABC_
ENTER	





(2) INPUT "文字", 変数, "文字", 変数………

INPUT "A=", A
INPUT "NAMAE=", B$, "TOSI=", C

データ待ちのとき、入力ガイダンス（入力案内）を表示させる方法です。

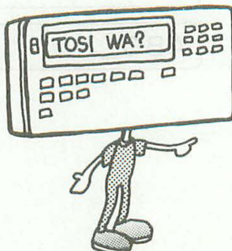
〈例〉
```
10 INPUT "A=", A
20 INPUT "NAMAE=", B$,
   "TOSI=", C
   ⋮
```

実行モード（RUN）

操作	表示
RUN ENTER	A=
12	12_
ENTER	NAMAE=
MARI	MARI_
ENTER	TOSI=
18	18_

このように、入力ガイダンスを表示させますと、計算機が何のデータを要求しているか、また、どのデータを入力すればよいかがわかりやすくなり、たいへん便利です。

なお、この形（"文字"と変数の間を，（コンマ）で区切った形）では、データを入れれば入力ガイダンスは消えます。





⑶　INPUT "文字"；変数，"文字"；変数…

INPUT "A="；A
INPUT "NAMAE="；B$，"TOSI="；C

この形（"文字"と変数の間を；（セミコロン）で区切った形）では、データを入れたとき入力ガイダンスは消えずに、その後に続けて入力した内容が表示されます。

〈例〉⑵の例で書き込んだプログラムを次のように変更して実行してみてください。

```
10  INPUT "A=";A
20  INPUT "NAMAE=";B$,
    "TOSI=";C
```

この部分を ,（コンマ）から；（セミコロン）に変えています。

実行モード（RUN）

RUN ENTER	A=_
12	A=12_
ENTER	NAMAE=_
MARI	NAMAE=MARI_
ENTER	TOSI=_
18	TOSI=18_

上記⑴、⑵、⑶を混合して使うこともできます。

INPUT　"A="；A，B，"C="，C

＝INPUT命令に続くステートメントのスキップ＝

プログラムがINPUT命令によるデータ入力待ちになっているとき、データなどの入力をおこなわずに ENTER キーを押しますと、いま実行しているところより後に書かれている命令などをスキップして（飛ばして）次のラインへ実行が移ります。

たとえば、次のような場合

```
    ⋮
40  INPUT  B, C:D=C*100
50
    ⋮
```

いま、Bへの入力待ちになっている場合、データを入力せずに ENTER キーだけを押しますと、Bへの入力命令も含めて、それ以降の命令（B，C：D=C*100）をスキップして50ラインへ実行が移ります。

同様に、Cへの入力待ちになっているときはC以降の命令（C：D＝C*100）がスキップされます。（ただし、変数はそれ以前のまま、内容が保持されています。）

2−2) 出力命令（PRINT）

PRINT命令は、計算などの処理結果やメッセージなど、指定された内容を表示部に表示させる命令です。
この命令は次の(1)〜(3)の形で用いることができます。

(1) PRINT { 式 / "文字" / 文字変数 }

この形では、式の値は表示部の右側にツメて表示し、文字および文字変数の内容は表示部の左側にツメて表示します。

PRINT 3	3.を右ヅメで表示します。
PRINT A+5	A+5の結果を右ヅメで表示します。
PRINT "ABC"	ABCを左ヅメで表示します。
PRINT B$	B$の内容を左ヅメで表示します。

〈例〉
```
10 A=8:B$="SHARP"
20 PRINT 12
30 PRINT A*6
40 PRINT B$
50 PRINT "EL-8"
```

実行モード(RUN)

RUN ENTER	12.
ENTER	48.
ENTER	SHARP
ENTER	EL−8

(2) PRINT { 式 / "文字" / 文字変数 } , { 式 / "文字" / 文字変数 }

16桁の表示部を左右8桁に分けて，の左右の内容を表示させます。このときも、8桁の範囲内で数値は右ヅメ、文字は左ヅメにします。

PRINT C,D	Cの値を左8桁内に、Dの値を右8桁内にそれぞれ右ヅメで表示します。
PRINT "A=",A	A=を左8桁内に左ヅメで、Aの値を右8桁内に右ヅメで表示します。

なお、表示内容が8桁を超える場合、文字は頭から8文字のみを表示し、数値は下位桁を切捨てて表示します。

ただし、数値が-10^6以下または10^6以上のときは浮動小数点方式（指数方式）で表示され、1より小さく−1より大きい数値は浮動小数点方式と固定小数点方式が自動的に選択され、表示されます。

(3) PRINT {式 / "文字" / 文字変数} ; {式 / "文字" / 文字変数} ; … ; {式 / "文字" / 文字変数}

PRINT "S=" ; S　左側にS=を表示し、それに続けてSの内容を表示します。

〈例〉
```
10 INPUT "SUUTI␣WA␣?␣"; A
20 B=A*A
30 PRINT "KOTAE=", B
40 PRINT "A="; A; "␣A*A="; B
50 END
```

␣：本機ではスペースを意味しています。計算機に入れるときは SPC キーを操作してください。

実行モード(RUN)

RUN ENTER	SUUTI WA ? _
80 ENTER	KOTAE=　　　6400.
ENTER	A=80.　A*A=6400.

式とは

本書において式とは、数値や数値変数およびこれらを含んだ計算式をさします。

(例)　3, A, 3*4, SIN B など

2−3）代入文（LET）

代入文は、変数に数値や文字を代入するための命令で、LET(レット)に続いて代入式を書きます。

LET　数値変数＝式

LET　文字変数＝"文字"（あるいは文字変数）

なお、代入文は通常LETを省略して使用できます。ただし、あとで説明しますIF文に続いて書く場合は、LETの省略はできません。

代入式は一般の数学とは違い、右辺の数値あるいは文字を左辺の変数に入れる（代入）することを意味します。したがって左辺は変数に限ります。

A=A+2	（Aの内容に2を加えてAに入れなさい）
B$="KOTAE"	（B$にKOTAEという文字を入れなさい）

2−4）終了文（END）

END（エンド）はプログラムの実行を終了させる命令です。

プログラムの終りには"これでプログラムは終了です！"という意味でEND文を書きます。

2−5）ジャンプ命令（GOTO）

ジャンプは"飛ぶ"ということで、プログラムの実行を指定したラインへ飛ばすことです。

通常プログラムは若いラインから順次実行されますが、GOTO（ゴートゥー）命令を実行しますと、GOTOに続いて書かれているラインへジャンプして、ジャンプしたラインから実行します。

この命令は次の形で用いることができます。

GOTO { 式 / "文字" / 文字変数 }

GOTO　40	40ラインへジャンプしなさい。
GOTO　12*5	60ラインへジャンプしなさい。
GOTO　A	Aの内容で示すラインへジャンプしなさい。 （Aの内容が50ならば50ラインへジャンプ）
GOTO　"PRT"	ラベル"PRT"のついているラインへジャンプしなさい。

〈例1〉

```
10  INPUT  "HANKEI=" ;R
20  S=π*R*R
30  PRINT  "MENSEKI=" ;S
40  GOTO  10
```

40ラインから10ラインへジャンプします。

〈例2〉

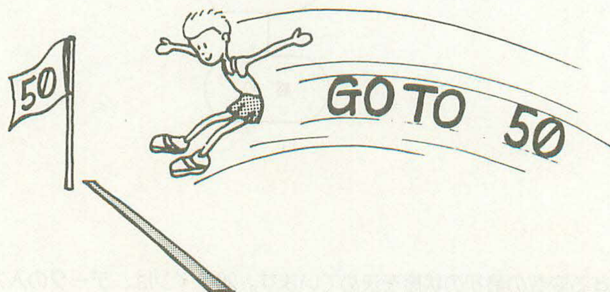
```
10  "A" : INPUT "HANKEI="
    ;R
20  S=π*R*R
30  PRINT "MENSEKI=" ;S
40  GOTO "A"
```

40ラインからラベル"A"へジャンプします。

このようなジャンプをラベルジャンプといいます。

ラベル：ラインの先頭につけた名前のことで、GOTO命令などでラベルジャンプが指定されていますと、実行の際、そのラベルを探してそのラインにジャンプします。ラベルジャンプの場合、プログラムの訂正などでラインナンバーをつけかえても、プログラムは変更する必要がないので便利です。
なお、同じラベルが2個以上書かれているときは、ラインナンバーの小さいほうへジャンプします。

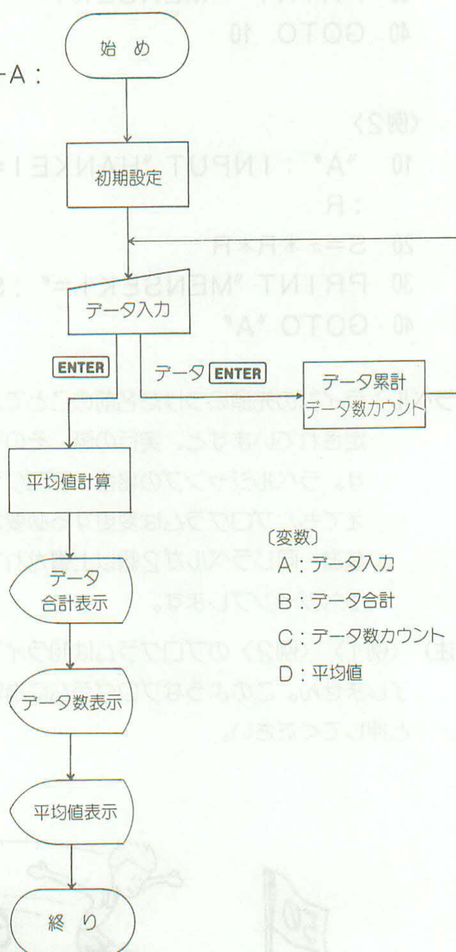
注）〈例1〉〈例2〉のプログラムは10ラインから40ラインまでをくり返し、実行が終了しません。このようなプログラムにおいて実行を終了させるときは ON BRK 、C·CE と押してください。



平均値を求める

これまで説明しました命令を使って平均値を求めるプログラムを作ってみました。

```
10 A=0:B=0:C=0
20 INPUT "DATA";A:B=B+A:
   C=C+1: GOTO 20
30 D=B/C
40 PRINT "GOUKEI", B
50 PRINT "DATA SUU", C
60 PRINT "HEIKIN", D
70 END
```



〔説明〕

10ラインは使用する変数の最初の状態を決めています。20ラインは、データの入力と、データの累計、データ数のカウントをしています。

入力するデータがある間は、「データ入力 ENTER 」の操作でくり返し20ラインを実行し、データがなくなれば ENTER キーのみを押して、20ラインを抜けます。(97ページのステートメントのスキップを参照)

そして30ラインで平均値を計算し、40〜60ラインでデータの合計、データ数、平均値を表示させています。

2−6）判 断（ＩＦ…ＴＨＥＮ～）

ＩＦ…ＴＨＥＮ～（イフ…ゼン～）は“もし…ならば～”という意味で、いろいろな判断をおこなう命令です。たとえば

ＩＦ Ａ＞＝１０ ＴＨＥＮ ＰＲＩＮＴ Ａ

とした場合、Ａの値が10以上（ＩＦ文が成立）ならばＩＦ文の後に続くＰＲＩＮＴ命令を実行し、10未満（ＩＦ文が不成立）ならば次のラインに実行が移ります。

ＩＦ文は通常、次の形で用います。

ＩＦ　条件式　ＴＨＥＮ　実行文

ただし、ＴＨＥＮを省略して

ＩＦ　条件式　実行文

という形でも用いることができます。

ＩＦ Ａ＞１０ ＴＨＥＮ ５０ [注1)]

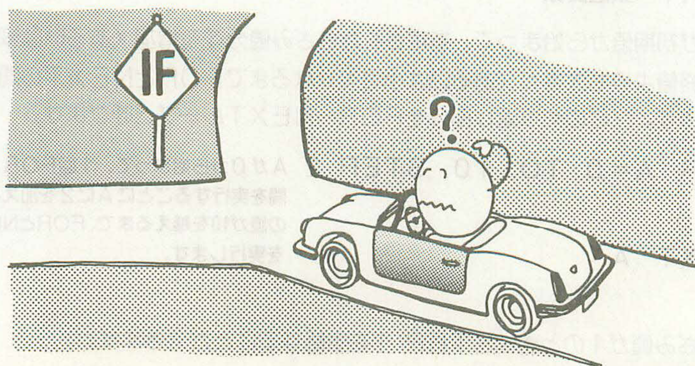
Ａが10より大きいときは50ラインへジャンプ。
10よりも小さいか、等しいときは次のラインを実行

ＩＦ Ｂ＜＝５ ＬＥＴ [注2)] Ｂ＝Ｂ＋１：ＧＯＴＯ ５０

Ｂが５より小さいか等しいとき、Ｂに１を加えて、50ラインへジャンプ。Ｂが５よりも大きいときは次のラインを実行

注１）…ＴＨＥＮ ５０は…ＴＨＥＮ ＧＯＴＯ ５０の形からＧＯＴＯを省略したものです。
ＴＨＥＮ を省略する場合は、ＧＯＴＯを省略することはできません。

注２）ＩＦ文に続いて代入文を書く場合は、必らずＬＥＴ命令が必要です。



ＩＦに続くＡ>10などの式を条件式といいます。条件式は通常次のような形で書きます。

条件式	判断内容	
○○ ＝ ××	等しいかどうか判断	(○○は××に等しいか？)
○○ ＞ ××	大きいかどうか判断	(○○は××より大きいか？)
○○>=××	以上かどうか判断	(○○は××以上か？)
○○ ＜ ××	小さいかどうか判断	(○○は××より小さいか？)
○○<=××	以下かどうか判断	(○○は××以下か？)
○○<>××	等しくないかどうか判断	(○○と××は等しくないか？)

(注)○○および××は式、変数を表わします。(5*4、A、8など)

なお、条件式をＡやＢ＋４のように変数や、通常の式のみにすることもできます。

```
IF B+4 THEN 6Ø
IF A THEN 6Ø
```

この場合は、変数や式の値が正（ＩＦ文が成立）ならばＩＦ文の後に続く内容を実行し、Øおよび負（ＩＦ文が不成立）ならば次のラインを実行します。

2−7）くり返し(FOR…TO…STEP, NEXT)

同じ計算や処理を何回もくり返しておこなうような場合は、ＦＯＲ−ＮＥＸＴ（フォー、ネクスト）文が便利です。
ＦＯＲ−ＮＥＸＴ文は次の形で用いられます。

```
FOR 数値変数＝初期値 TO 最終値 STEP きざみ値
    ∫
NEXT 数値変数
```

数値変数が初期値から始まって、指定されたきざみ値分ずつ増加（あるいは減少）していき、最終値よりも大きく（あるいは小さく）なるまでＦＯＲとＮＥＸＴの間をくり返し実行します。（このくり返し部分をＦＯＲ−ＮＥＸＴループと呼びます。）

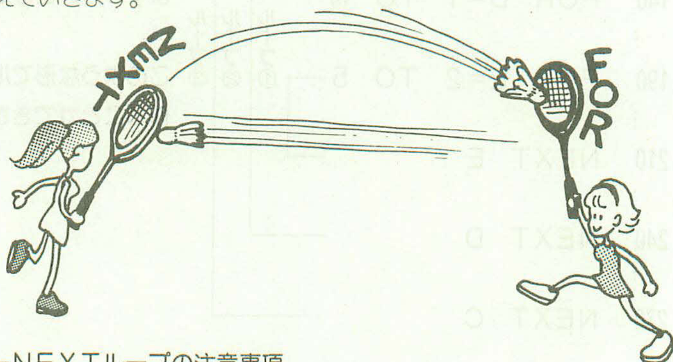
```
FOR A=Ø TO 1Ø STEP 2
    ∫
NEXT A
```

ＡがØから始まって、１回ＦＯＲとＮＥＸＴ間を実行するごとにＡに２を加えながら、Ａの値が10を越えるまで、ＦＯＲとＮＥＸＴの間を実行します。

なお、きざみ値が１のときはＳＴＥＰ １を省略することができます。

〈例〉
```
10 B=0
20 FOR A=1 TO 100
30 B=B+A
40 NEXT A
50 PRINT "1+2+.. +100=";B
60 END
```
(20〜40: FOR−NEXT ループ)

●このプログラムは1から100までの数値を累計するプログラムです。20〜40ラインがくり返しループであり、Aを1から100まで変化させ、それをBに加わえていきます。



⊙ FOR−NEXTループの注意事項

FOR−NEXTループを使う場合に注意すべきことがいくつかありますので、次に説明します。

① FORとNEXTは必ず対にして使い、FORの後の数値変数とNEXTの後の数値変数は同一でなければなりません。

```
FOR B=1 TO 5
   ⟆
NEXT B
```
同じ数値変数にする。

② 初期値、最終値、きざみ値（ステップ数）は次の範囲内で指定できます。

−9.999999999E99〜9.999999999E99

$(-9.999999999\times10^{99}\sim9.999999999\times10^{99})$

ただし、次のように初期値にきざみ値を加えると、最終値から離れていってしまう場合は、ループ内を１回だけ実行してループを抜けます。

```
FOR A=1 TO 5 STEP -2
   ⟆
NEXT A
```

なお、きざみ値を0に指定しますと、永遠にループ内の実行をくり返すプログラムになってしまいます。

③ FOR−NEXTループの中に、別のFOR−NEXTループを入れることができます。

ただし、中に入るFOR−NEXTループは外のFOR−NEXTループ内に完全に入っていなければなりません。この条件でループを5段まで重ねて使う（深みをもたせる）ことができます。

（例1）

```
110  FOR C=1  TO 12 ──────────┐
 :                              │
140  FOR D=1  TO 10 ─────┐     │
 :                         │     │
190  FOR E=2  TO 5 ─┐     │     │
 :                    │     │     │
210  NEXT E ─────────┘     │     │
 :                  ループ① ループ② ループ③
240  NEXT D ───────────────┘     │
 :                                │
270  NEXT C ─────────────────────┘
```

それぞれ、中のループが外のループの中に完全に入っていますので、この使い方は正しい。

このような形でループ⑤まで作ることができます。

（例2）

```
10 FOR C=1  TO 10 ───────┐
 :                         │
40 FOR D=1  TO 12 ─┐     │
 :                   │     │
90 NEXT C ──────────│─────┘
 :                   │
110 NEXT D ─────────┘
        ループ①  ループ②
```

ループ①とループ②が交差していますのでこの使い方は誤り（ERROR 5）

③ 外からFOR−NEXTループ内に飛び込むことはできません。

（例）

```
 :
40  GOTO 80 ──────────────┐
 :                          │
60  FOR H=1  TO 10 ─┐     │ 飛び込み
 :                    │     │
80  A=C*C+10  ←──────│─────┘
 :                    │
100 NEXT H ──────────┘
```

ループの外からループ内に飛び込むことはできません。（ERROR 5）

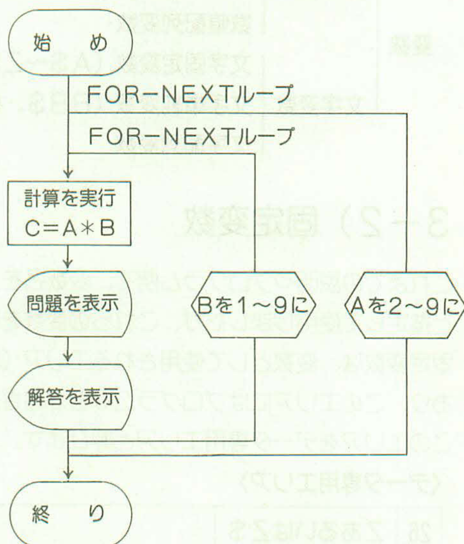
注）FOR−NEXTループから外に抜け出すようなプログラムでは、FOR−NEXTループが終了したことにならず、FOR文が何回か実行されるうちにエラー（ERROR　5）になります。
したがって、FOR−NEXTループを途中で抜ける場合は、カウント用の数値変数に最終的を代入したうえで、NEXT文の書かれているラインを実行させることにより、FOR−NEXTループを終了させてください。

（例）
```
      :
100  FOR A=Ø TO 30
      :
150  IF B=50 LET A=30:GOTO 200
      :
200  NEXT A
```
Bが50になったらFOR−NEXTループの実行を中止します。

九九の練習

それでは、FOR−NEXTを使って九九の練習をおこなうためのプログラムを作ってみましょう。

```
10 FOR A=2 TO 9
20 FOR B=1 TO 9
30 C=A*B
40 PRINT "␣␣␣␣";A;"X␣";
   B;"=␣"
50 PRINT "␣␣␣␣";A;"X␣";
   B;"=␣";C
60 NEXT B
70 NEXT A
80 END
```



このプログラムは、 ENTER を押していくことにより、問題を出して次にその答を出していくプログラムです。
FOR−NEXTループを2段にして使い、A*Bの式において、はじめにAに2を入れ、Bを1から9まで変化させて、2の項の問題と解答を出します。
次にAに3を入れ、またBを1から9まで変えて、3の項の問題と解答を出します。
このようにして、9の項までの問題と解答を出していきます。

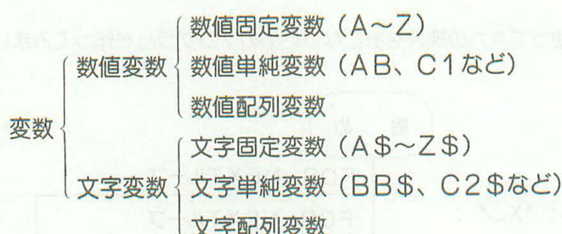
（このプログラムでは、外側のループ（10〜70ライン）を1回実行すると内側のループ（20〜60ライン）は9回実行されます。
外側のループは変数を2〜9まで変えて実行しますので8回実行されることになります。
この場合、内側のループは72回も実行されることになります。）

3. 変数

これまで説明しました命令などの機能は、ご理解いただけましたでしょうか？
これらの機能が使いこなせれば、もう本機のＢＡＳＩＣは70％理解できたといってもよいでしょう。
この項では変数とメモリーについてくわしく説明します。
プログラムを作るうえで変数の使いかたが大切になってきますので、十分理解するようにしてください。

3－1）変数の構成

変数には数値変数と文字変数があることは68ページで述べましたが、これらにも次のような種類があります。



3－2）固定変数

これまでの説明やプログラム例で、変数名をＡ、Ｂ、Ｃ…あるいはＡ＄、Ｂ＄、Ｃ＄…と指定して使用しましたが、これらの変数を固定変数と呼びます。
固定変数は、変数として使用されるエリア（区域）がメモリー上に独立して確保されており、このエリアにはプログラムなどが書き込まれることはありません。
このエリアをデータ専用エリアと呼びます。

〈データ専用エリア〉

26	ＺあるいはＺ＄	
⋮	⋮	⋮
3	ＣあるいはＣ＄	
2	ＢあるいはＢ＄	
1	ＡあるいはＡ＄	
	変数名	データの記憶場所

この図で示しますように、このエリアは26に区切られており、変数A、変数B…として使用される場所はすべて決っています。
また、同じ名前の数値固定変数と文字固定変数は同じ場所が使用されます。

◉ **数値固定変数：A～Zの変数**

仮数部10桁、指数部－99～99で表わすことのできる数値を扱います。

◉ **文字固定変数：A＄～Z＄の変数**

文字固定変数は最大７字までの文字・記号を扱います。

３－３）単純変数

単純変数は、変数名をＡＡやＢ１のように２文字（あるいはそれ以上）で指定する変数です。
この変数は固定変数のように、使用される場所は決っておらず、初めて使用したときにメモリー内（プログラム・データエリア内）に自動的に確保されます。
また、同じ変数名でも数値単純変数と文字単純変数は別べつに確保されますので、たとえばＡＢとＡＢ＄を同時に使用することができます。
なお、単純変数の変数名には英文字と数字を用いることができますが、最初の文字は英文字でなければなりません。
そして、変数名に３文字以上を用いた場合は前２文字のみが有効になります。

注）本機が持っている関数命令やＢＡＳＩＣ命令は変数名として使用することはできません。

例）ＰＩ、ＩＦ、ＴＯ、ＯＮ、ＳＩＮ……などは変数名として使用できません。

◉ 文字単純変数は最大16字までの文字・記号を扱います。

3－4）配列変数

数学などで、同じ性質の複数個のデータを表わすとき、たとえば

x_1、x_2、x_3……x_n

のように、1つの変数名に添字をつけて表わす場合があります。
BASICでも同様に、1つの変数名に（　）で囲った添字をつけて、次のように表わすことができます。

X(1)、X(2)、X(3)……X(10)

このように添字をつけて表わす変数を配列変数と呼びます。

(1) DIM

配列変数を使用するときは、事前にDIM（ディメンジョン）命令によって配列名とその大きさを定義（宣言）しておかなければなりません。
DIM命令で定義をすることによって、その大きさの配列変数がメモリー（プログラム・データエリア）上に確保されます。たとえば

```
DIM  X(5)
```

は、次の図のように配列名Xの6個の変数（配列要素）を確保することになります。
（X(Ø)からX(5)までの6個）

X(Ø)	X(1)	X(2)	X(3)	X(4)	X(5)

複数の配列名を使用する場合は、DIM命令で1度に定義することができます。

```
DIM  X(5), Y(10), Z$(15), AB(3)
```

添字は理論的に0～255までの整数値を用いることができます。

（ただし、計算機のメモリーの大きさ、および使用状態によっては添字で指定しただけ変数が確保できない場合があります。）

もし、添字が小数部を含んでいるときは小数部は無視され、整数部のみが有効になります。

X(2.4)　→　X(2)とみなします。

Y(Ø.25)　→　Y(Ø)とみなします。

なお、添字は数値変数や数式の形で用いることもできます。

```
10  INPUT  A, B
20  DIM  X(A), Y(B-1)
```

⑵　一次元配列

添字が１個だけのものを一次元配列といいます。(リスト変数ともいいます。)
配列変数はデータの集計などに便利に使える変数です。
たとえば、次の表を見てください。

（　）内の数字は項目№です。

科目／名前	国語(1)	数学(2)	英語(3)	物理(4)	化学(5)	合計(6)	平均(7)
M	95	70	87	60	68		

これはMさんの試験の結果を表にしたものです。
配列変数を用いますと、この表と同じような考えかたで計算機に記憶させることができます。
たとえば、次のプログラムを実行し、データ（各科目の点数）を入力しますと、下の表のようにデータが記憶され、合計、平均が記憶されます。

```
10  DIM  T(7)
20  FOR  A=1  TO  5
30  INPUT  "DATA=" ;T(A)
40  NEXT  A
50  FOR  A=1  TO  5
60  T(6)=T(6)+T(A)
70  NEXT  A
80  T(7)=T(6)/5
100 INPUT  "KAKUNIN␣
    NO. =?␣" ;A
110 PRINT "NO. " ;A; "␣␣␣" ;
    T(A); "TEN"
120 GOTO  100
```

- 配列の定義（10）
- データの入力（各科目の点数を入れます）（20〜40）
- 合計を求める（50〜70）
- 平均を求める（80）
- 呼び出したい項目の№を入れます（1〜7）（100）
- 指定した項目の点数を表示します。（110）

T(Ø)	T(1)	T(2)	T(3)	T(4)	T(5)	T(6)	T(7)
	95	70	87	60	68	380	76

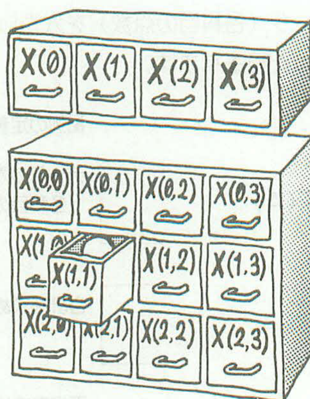
このプログラムで、20〜40ラインを見てください。
FOR−NEXTループを使ってデータの入力をおこなうところです。
ここで30ラインのＩＮＰＵＴ文に配列変数が使われていますがＴ(Ａ)のように添字が変数になっています。
このようにしておきますとFOR−NEXT文によりAが１、２……５と変化するにつれて配列Ｔの指定もＴ⑴、Ｔ⑵…Ｔ⑸と変化します。
したがって、同じＩＮＰＵＴ文でＴ⑴、Ｔ⑵…Ｔ⑸に順番にデータを入れていくことができます。

同様に60ラインでもT(A)が使われています。

このラインはT(6)にT(A)すなわちT(1)、T(2)…T(5)の内容を加えていきます。110ラインのT(A)の意味はわかりますね。100ラインのINPUT文でAに項目のナンバー、たとえば6を入力すれば、110ラインのT(A)がT(6)と指定されたことになり、その内容380(合計)が表示されます。

このように配列変数は添字を変数にすることができるため、同じプログラムで自由に配列要素(配列の中の個々の変数)が指定できるため、データがたくさんあるときはたいへん便利です。

なお、このプログラムではT(Ø)を使用していません。これは、この例ではØという項目がないため、T(Ø)からデータを入れるとナンバーがずれて、使いにくいからです。メモリーに余裕がある場合は、このようにすることもよくあります。



(3) 二次元配列

本機は添字を2個まで使用できます。

添字が2個のものを二次元配列といいます。(テーブル変数ともいわれます。)

一次元配列が、いうなれば横一列(あるいは縦一列)の配列だったのに対し、二次元配列は縦と横の配列といえます。

下の図を見てください。これは

DIM X(3, 5)

と定義したときに確保される変数(配列要素)を表わしたものです。

X(0, 0)	X(0, 1)	X(0, 2)	X(0, 3)	X(0, 4)	X(0, 5)
X(1, 0)	X(1, 1)	X(1, 2)	X(1, 3)	X(1, 4)	X(1, 5)
X(2, 0)	X(2, 1)	X(2, 2)	X(2, 3)	X(2, 4)	X(2, 5)
X(3, 0)	X(3, 1)	X(3, 2)	X(3, 3)	X(3, 4)	X(3, 5)

これは、DIM X(5)と定義したときの図(110ページ)と見比べていただければ、おわかりのように、一次元の配列を4段重ねた形ですね。

したがって変数(配列要素)は、横に6個(5+1個)のものが縦に4段(3+1段)重なっていますので、24個確保されていることになります。

それでは、一次元配列で用いましたMさんの試験結果の表に、NさんとOさんの表を重ねて、各科目ごとの3人の合計と平均も求められるプログラムを作ってみましょう。

科目 / 名前	国語(1)	数学(2)	英語(3)	物理(4)	化学(5)	合計(6)	平均(7)
M(1)	95	70	87	60	68		
N(2)	82	75	93	68	65		
O(3)	100	90	87	72	80		
合計(4)							
平均(5)							

```
10 DIM T(5, 7)
20 FOR B=1 TO 3
30 PRINT B; "RETU␣ME␣IRERU"
40 FOR A=1 TO 5
50 INPUT "DATA=" ;T(B, A)
60 NEXT A
70 NEXT B
```

配列を定義します。（10）

入力ループ（20〜70）
50ラインのINPUT命令のT（B，A）において、Bで縦の項目を指定し、Aで横の項目を指定します。FOR－NEXTループを2重に使い、Bが1のときAを1〜5まで変えながらMさんの点数を入れ、Bが2のとき、またAを1〜5まで変えながらNさんの点数を入れます。

```
100 FOR B=1 TO 3
110 FOR A=1 TO 5
120 T(B, 6)=T(B, 6)+T(B, A)
130 T(4, A)=T(4, A)+T(B, A)
140 NEXT A
150 NEXT B
```

合計計算ループ
縦の合計と横の合計を求めています。
120ラインは各人の合計
130ラインは各科目の合計

```
200 FOR A=1 TO 5
210 T(5, A)=T(4, A)/3
220 NEXT A
250 FOR B=1 TO 3
260 T(B, 7)=T(B, 6)/5
270 NEXT B
```

各科目の平均点を求めます（200〜220）

各人の平均点を求めます。（250〜270）

```
300 INPUT "KAKUNIN␣
    NO.B=?␣" ;B
310 INPUT "NO.A=?␣" ;A
320 PRINT "NO." ;B;A;"␣␣" ;
    T(B, A) ; "TEN"
330 GOTO 300
```

縦と横の番号を指定して、データや結果を表示させます。Bが縦の番号、Aが横の番号です。

T(0,0)	T(0,1)	T(0,2)	T(0,3)	T(0,4)	T(0,5)	T(0,6)	T(0,7)
T(1,0)	T(1,1)	T(1,2)	T(1,3)	T(1,4)	T(1,5)	T(1,6)	T(1,7)
	95	70	87	60	68	380	76.0
T(2,0)	T(2,1)	T(2,2)	T(2,3)	T(2,4)	T(2,5)	T(2,6)	T(2,7)
	82	75	93	68	65	383	76.6
T(3,0)	T(3,1)	T(3,2)	T(3,3)	T(3,4)	T(3,5)	T(3,6)	T(3,7)
	100	90	87	72	80	429	85.8
T(4,0)	T(4,1)	T(4,2)	T(4,3)	T(4,4)	T(4,5)	T(4,6)	T(4,7)
	277	235	267	200	213		
T(5,0)	T(5,1)	T(5,2)	T(5,3)	T(5,4)	T(5,5)	T(5,6)	T(5,7)
	92.3…	78.3…	89.0	66.6…	71.0		

前記プログラムでは図の太い線で囲った変数のみを使っています。

このプログラムで20～70ラインがデータ入力のループです。
FOR－NEXTループを2段に使って、T（B，A）のBとAの値を変えて、指定する変数を変えています。
たとえば、はじめにBに1を入れ、Aの値を1から5まで変化させることによってT（1，1）からT（1，5）までを指定し、次にBに2を入れて、またAの値を1から5まで変化させれば、T（2，1）からT（2，5）の変数が指定されます。
このように二次元配列変数は2重のFOR－NEXTループと組み合わせて使えば、データがいくらあっても、ほとんど同じプログラムの形で使用でき、たいへんに便利です。

(4) 文字配列変数

配列変数にも文字変数があり、配列名に＄（ドル）記号をつけて表わします。

```
DIM Z$(9)          DIM X1$(2, 1)
DIM Y$(5, 4)       X1$(0, 1)="NAKAMURA"
```

◎ 文字配列変数の拡張

文字固定変数は、1個の変数に最大7文字、文字単純変数は1個の変数に最大16文字記憶できる固定長の変数でしたが、文字配列変数は1～80文字の範囲で、任意に変数の長さを指定することができます。

```
DIM C$(9)*30
```

C$(0)～C$(9)の各変数には、それぞれ最大30文字まで記憶することができます。

```
DIM N$(5, 4)*6
```

N$(0, 0)～N$(5, 4)の各変数には、それぞれ最大6文字まで記憶することができます。

このように、DIM文に ＊式 をつけて、変数の長さを指定します。
なお、長さを指定しない(＊式 がない)ときは自動的に16文字が指定されます。

(5) 配列変数を使うときの注意点

① 同一プログラム内で、同じ配列名を再定義することはできません。
たとえば、ＤＩＭ　Ｘ(５)とＤＩＭ　Ｘ(３，４)は同じＸという配列名になります。
もし、定義し直したい場合はＣＬＥＡＲ命令(118ページ参照)により、変数を消去してから、定義し直してください。
なお、数値配列変数と文字配列変数は別べつに確保されますので、たとえば配列Ｚと配列Ｚ＄は同時に使用することができます。

② 配列変数の添字として、文字列を用いる関数(ＡＳＣ、ＶＡＬ、ＬＥＮ)を使用することはできません。

ＤＩＭ　Ｂ(ＶＡＬ"１２")

のような形で使用しますとエラーになります。

③ ＲＵＮ命令によりプログラムを実行したとき、それまでにメモリー上に確保されていた配列変数は消去されます。(129ページを参照)

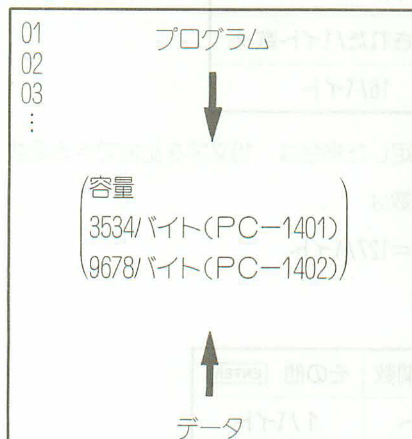
ＧＯＴＯ命令や定義付けキーによるプログラムの実行では、変数は消去されません。
したがって一度実行したプログラムをＧＯＴＯ命令などで再実行させるとき、ＤＩＭ命令の書かれているラインを実行させると、同じ変数名を再定義することになり、エラー(ＥＲＲＯＲ　３)になります。

④ 配列Ａを使用するときは特別な注意を必要とする場合がありますので、119ページをご参照ください。

3−5）メモリーの構成と変数

本機のメモリーの中で、通常プログラムやデータを記憶するために使用されるのはプログラム・データエリアとデータ専用エリアです。

〈プログラム・データエリア〉



プログラムがこちら側から書き込まれていきます。

C（1）やB（5，8）のような配列変数、及びAAやB1$のような単純変数（2文字変数）はこちら側から確保されます。

〈データ専用エリア〉（容量208バイト）

26	ZあるいはZ$	
25	YあるいはY$	
⋮	⋮	
2	BあるいはB$	
1	AあるいはB$	

固定変数（26メモリー）

メモリーはこの図のような構成になっています。プログラム・データエリアはプログラムメモリー、データメモリー（配列変数、単純変数）のいずれにも使用することのできるメモリーです。図で示しますように、プログラムと変数はそれぞれ逆方向から書き込まれ、確保されていきます。これに対して専用エリアは常にデータメモリー（変数）として確保されています。通常、データメモリー（変数）として使用できるのはこの２種類のメモリーエリアですが、プログラム・データエリアを使用するときは書き込まれているプログラムの長さによって変数として使える大きさが変ってきますので注意が必要です。

次に各変数を確保する場合に使用するバイト数およびプログラムの各命令などの占めるバイト数を記しておきますので参考にしてください。

変　数	変数の名前	データ	
数値変数	7 バイト	8バイト	
文字変数	7 バイト	配列変数	指定されたバイト数※
		単純変数	16バイト

※たとえば　ＤＩＭ　Ｂ＄(２，３)＊10　と指定した場合は、10文字を記憶できる変数を12個確保しますが、これに使用するバイト数は

7バイト(変数名)＋10バイト(文字数)×12個＝127バイト

になります。

構成要素	ラインナンバー	命令文、関数	その他 ENTER
使用バイト数	3バイト	1バイト	1バイト

3－6）変数をクリア(消去)するには(ＣＬＥＡＲ)

⊙ 個々の変数をクリア(消去)するには次のようにします。

(例)　Ａ＝０
　　　Ｂ(1)＝０　} 数値変数は０を代入することによりクリアします。
　　　Ａ＄＝""
　　　Ｃ＄＝""　} 文字変数は""を代入することによりクリアします。

("" を代入するということは文字変数を文字がない状態にするということです。
この状態をＮＵＬ（ヌル）あるいはＮＵＬ状態と呼びます。(ＮＵＬ：ＮＵＬＬ))

⊙ ＣＬＥＡＲ命令(クリア)

すべての変数を一度にクリアするには、ＣＬＥＡＲ命令を実行します。ＣＬＥＡＲ命令の一般形は次のようになります。

```
CLEAR
```

この命令が実行されれば、ＤＩＭ命令により確保されていた配列変数や、単純変数は消去され、固定変数の内容もすべて消されます。

● この命令ではプログラムはクリアされず、保護されます。

3−7）変数A（ ）について

固定変数はデータ専用エリアを使用していますが、このデータ専用エリアを配列変数と同様の添字つき変数として指定することができます。

固定変数はA～Z（A＄～Z＄）の26本あります。これにそれぞれ1～26の番号を割り当て、A（1）～A(26)あるいはA＄（1）～A＄(26)とすることにより、添字つき変数として指定されます。すなわち、A（1）はAと同じであり、A（2）はBと同じ変数になります。

ただし、すでにDIM命令により配列Aあるいは配列A＄が定義されている場合は、固定変数を添字つきの変数として指定することはできません。たとえば

DIM A(5)

により配列Aが定義されているときは、プログラムデータエリアにA（Ø）～A（5）が確保されていますので、A（2）を指定した場合は固定変数（B）ではなく、プログラムデータエリアに確保されているA（2）を指定したことになります。

また、たとえばA（9）とした場合は配列の定義外の指定とみなされてエラー（ERROR 3）になります。

逆に、固定変数をA（ ）の形で使用した場合はDIM命令により配列A及び配列A＄を定義することはできなくなります。

ただし、CLEAR命令により変数をクリアすれば定義可能になります。

⊙添字を27以上にした場合

DIM命令により配列Aの定義がおこなわれていないときに、A（ ）の形で添字を27以上にした場合は、固定変数の延長として、プログラム・データエリアにその変数が確保されます。

たとえば A(35)=5 を実行すれば、プログラム・データエリアにはA(27) ～ A(35)の変数が確保されます。

A(27)以上の変数は配列変数として扱われますが、ほかの配列変数と違って次のような注意が必要です。

1．配列を分けることはできません。

次のように配列が分かれるような場合はエラーになります。

```
10  DIM  B(2)
20  A(28)=5
30  BC=12
40  A(30)=9
 ：
```

〈プログラム・データエリア〉

A(30)
A(29)
BC
A(28)
A(27)
B(2)
B(1)
B(Ø)

このプログラムを実行しますと、右図のような形で、変数が確保され、40ラインの実行でA(29)、A(30)を確保しようとしたとき、A( )の配列が2つに分かれるため、エラー（ERROR 3）になります。

2．同じ添字の数値変数と文字変数は共存できません。

たとえば、A(30)とA＄(30)は同じ場所を使用することになりますので、同時に使用することはできません。

3．二次元配列にしたり、格納文字数の指定はできません。

A＄( )は最大7文字まで記憶できる固定長の変数になります。

4．添字がØの変数は存在しません。A(Ø)やA＄(Ø)を指定するとエラーになります。

4. 定義付けプログラム

本機はプログラム・データエリアの容量内であれば、何本でもプログラムを書き込むことができます。
そして、プログラムが複数本書き込まれているときに、2番目以降のプログラムを実行する場合は、次の方法でスタートさせることができます。

① RUN　ラインナンバー　ENTER

② GOTO　ラインナンバー　ENTER

この方法により、指定したラインからプログラムの実行が開始されます。この方法では操作に手間がかかり、また、ラインナンバーを常に知っていなければならず、なかなかわずらわしいものです。このようなとき、プログラムラインの始めに "A" や "B" のようにラベルを書いておくことにより、DEF A や DEF B と押すだけでプログラムの実行を開始させることができます。
定義付けとは、プログラムを下に示します18のキー（定義付けキー）のどれかに割り当てることです。
これらのキーに対応する文字、記号をラベルとしてラインナンバーのすぐ後に書き込んでおけば、DEF A 、DEF S 、……と押したときに、そのキーに対応するラベルの書かれているラインからプログラムが実行されます。

（例）

```
      :   ┌ラベルA
  50  : "A" : INPUT  A
      :   ┌ラベルS
 120  : "S" : N=0 : T=0
      :   ┌ラベルD
 210  : "D" : CLEAR
      :
```

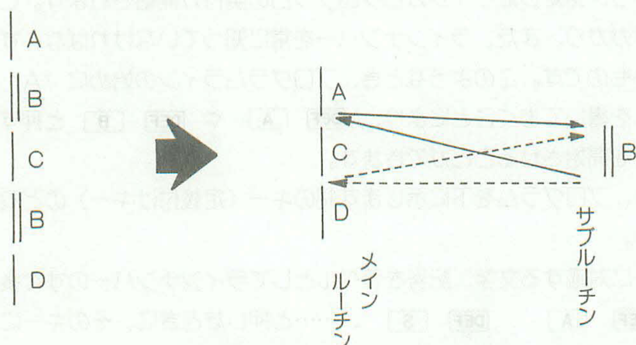
● 定義付けキー

定義付けのできるキーは次の18キーです。

A　S　D　F　G　H　J　K　L　,
Z　X　C　V　B　N　M　SPC

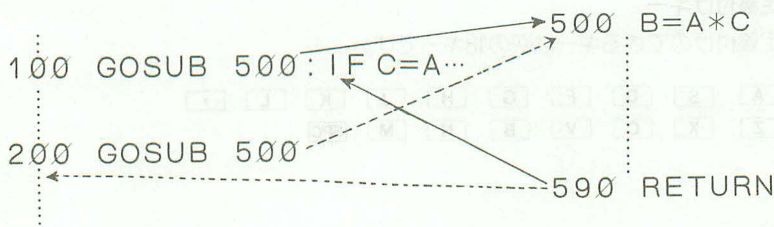
5. サブルーチン(GOSUB、RETURN)

ある一連のプログラムで、何回も同じ計算や処理が出てくる場合に、その部分を抜き出してプログラムしておき、必要に応じてその抜き出したプログラムを実行しに行くようにすればプログラムを短かく、また簡略化することができます。
たとえば、次の左図のように共通部(B)があるとき、その部分を抜き出すと(サブルーチンにすると)、右図のようになります。

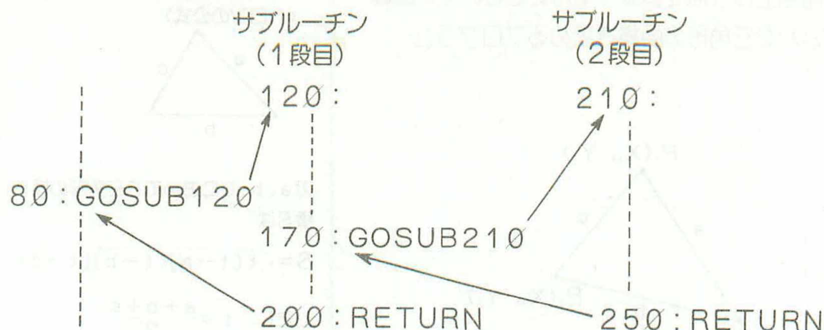


サブルーチンへのジャンプは、GOSUB命令(ゴーサブ)によりおこないます。
サブルーチンのおかれているラインナンバー(あるいはラベル)をGOSUBの後に続いて書いて、サブルーチンジャンプを指示します。
また、サブルーチンの最後にはRETURN命令(リターン)を書いて、メインルーチンへの復帰を指示します。
メインルーチンへ復帰したときは、GOSUBのあった次の命令を引続き実行します。

〈例〉



●サブルーチンの中にサブルーチンを入れることもできます。この場合、10段まで深みをもたせることができます。



●GOSUB命令の一般形は次のようになります。

GOSUB 式

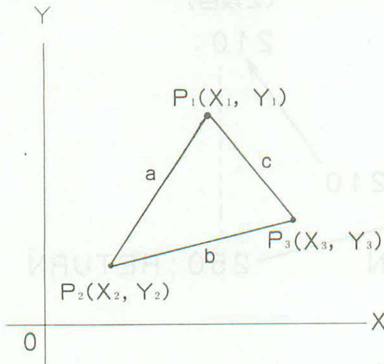
あるいは

GOSUB "文字"

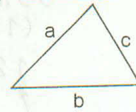
GOSUB命令の指定の形はGOTO命令と同様です。

〈プログラム例〉

直交座標上の３点を直線で結んだとき、その直線で囲まれた三角形の面積を求めるプログラム



〈ヘロンの公式〉

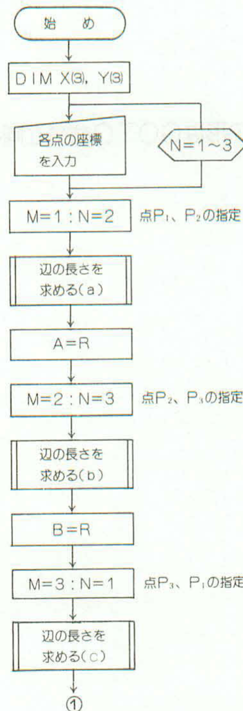


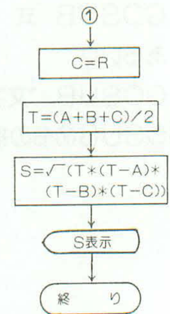
辺a、b、cにおいて、三角形の面積Sは

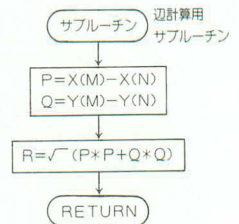
$$S=\sqrt{t(t-a)(t-b)(t-c)}$$

ただし、$t=\frac{a+b+c}{2}$

```
 10:DIM X(3),Y(3)
 20:FOR N=1 TO 3
 30:INPUT "X=";X(N),"Y="
    ;Y(N)
 40:NEXT N
 50:M=1:N=2: GOSUB 200
 60:A=R
 70:M=2:N=3: GOSUB 200
 80:B=R
 90:M=3:N=1: GOSUB 200
100:C=R
110:T=(A+B+C)/2
120:S=√(T*(T-A)*(T-B)*(T
    -C))
130:PRINT USING "#####.#
    #";"S=";S
140:END
200:REM HEN NO NAGASA
210:P=X(M)-X(N)
220:Q=Y(M)-Y(N)
230:R=√(P*P+Q*Q)
240:RETURN
```







プログラムスタート後、各点のX座標、Y座標の値を入力すれば、面積Sが求められます。プログラムでは、各点の座標から、辺a、b、cの長さを計算し、ヘロンの公式より、面積Sを求めています。

なお、各辺を計算するプログラムをサブルーチンとして200ライン以降に書いています。

注）200ラインはREM文といって、プログラムチェックをおこなうときなどに、わかりやすくするために注釈を入れる命令で、プログラム実行時は無視されます。（133ページ参照）

6．表示に関するBASIC

6－1）フォーマット指定（USING）

USING命令（ユージング）は数値などを表示するときに、その表示フォーマットを指定します。

○USING命令の一般形

①	USING	"###"	符号を含めて整数3桁を表示
②	USING	"###."	符号を含めて整数3桁と小数点を表示
③	USING	"###.##"	符号を含めて整数3桁と小数点と小数点以下2桁表示
④	USING	"##.##^"	小数以下2桁までの指数方式での表示（このとき、仮数部の整数は符号を含めて2桁、指数部は符号を含めて4桁（E－○○）が自動的にとられます。）
⑤	USING	"&&&&&&"	文字を6桁表示
⑥	USING	"###&&&&"	数値と文字を同時に指定
⑦	USING		フォーマット指定を解除

○USING命令に続く"　　"内の記号はそれぞれ次の指定をおこないます。

#	数値の桁数を指定 整数部桁数：2～11桁（符号を含む） 指定桁より数値のほうが少ないときはその分だけスペース表示になり、多いときはエラーになります。 12桁以上指定した場合はすべて11桁の指定とみなされます。 小数部桁数：0～12桁（浮動小数点表示のときは0～9桁） 指定桁より数値のほうが少ないときはその分だけ0表示になり多いときはその分だけ切捨てられます。
.	小数点の表示を指定（整数部と小数部の区切りを指定）
^	数値の浮動小数点表示（指数方式）を指定 整数部の指定桁にかかわりなく、仮数部の整数桁は2桁（符号を含む）になります。 小数部の桁数を9桁以上に指定した場合、仮数部の小数桁は9桁になります。
&	文字列の桁数を指定 指定桁より文字数が少ない場合はその分スペース表示になり、文字数が多い場合は指定された桁分だけ表示します。

○USING文を単独に使います。

```
10:A=12.3,B=-34.5678
20:USING "####.###"
30:PRINT A;B
40:END
```

表示例

```
 12.300 -34.567
```

␣スペース ␣スペース ␣スペース

○PRINT文の中にUSING文を含め、PRINT USING文として使います。

```
10:B=-10,C=10.7703
20:PRINT USING "&&&###"
   ;"B";B;" C="; USING
   "###.###";C
30:END
```

表示例

```
B  -10 C= 10.770
```

␣スペース ␣スペース ␣スペース ␣スペース

○USINGの解除

USING

USINGで表示フォーマットの指定をおこないますと、以降に実行されるPRINT命令、PAUSE命令には、すべてそのフォーマット指定が有効になりますので、不必要なときは本命令によってフォーマット指定を解除しておきます。

```
10:A=34,B=7
20:C=A/B
30:USING "&&&###"
40:PRINT "A=";A
50:PRINT "B=";B
60:PRINT USING ;"C=";C
70:END
```

```
10:A=34,B=7
20:C=A/B
30:USING "&&&###"
40:PRINT "A=";A
50:PRINT "B=";B
60:USING
70:PRINT "C=";C
80:END
```

60ラインのUSINGによりフォーマット指定を解除しています。

USING文を使用する場合のご注意

USING命令により、表示フォーマットが指定されている場合、PRINT命令の形によっては指定されたフォーマットで表示できなくなり、エラー（ERROR 7）になります。（185ページを参照ください。）

① 「PRINT 式」の場合は、USINGで指定された総桁数が16桁を越えているとエラーになります。
ただし、整数部の指定桁が11桁を越えているときは、越えた部分が無視されます。

```
                      16桁
10 USING "###############.####"
                   11桁
11桁を越えた部分は無視されます。
```

この指定の場合、総桁数は16桁とみなされるため、エラーになりません。

② 「PRINT 式，式」の場合は、USINGにより固定小数点方式が指定されているとき、整数部（符号、小数点を含む）の桁数が8桁を越えているとエラーになります。

③ 「PRINT 式；式」の場合は、エラーになりませんが、表示される桁は左から16桁目までです。

なお、10進→16進変換の結果（16進数値）を表示するとき、USINGにより浮動小数点表示方式（指数表示）が指定されている場合はエラー（ERROR 7）になります。

6−2）表示時間の指定（WAIT）

WAIT命令（ウエイト）は、PRINT命令によるプログラムの停止時間を指定する命令です。WAIT命令による指定はPRINT命令のすべての形に対して有効で、プログラムは指定された時間だけ停止し、その後、自動的に実行を再開します。

○停止時間は次のように指定します。

WAIT 式

式の値により、決められている時間だけプログラム実行を停止して内容を表示し、その後プログラムの実行を再開します。

式の値は0〜65535まで指定できます。なお、式の値1は約1/59秒に相当します。

○WAIT指定の解除

WAIT

WAITでの指定は、以降のPRINT命令に対してすべて有効となるため、時間を無限にしたいときなどは、本命令によって時間指定を解除してください。

```
 10:DIM B$(0)*20,E$(0)*2   ← 配列変数の定義
    0
 20:WAIT 20                ← 停止時間を指定
 30:INPUT "NAMAE_WA_?_",   ← 名前を入力してください。
    B$(0)
 40:B= LEN B$(0)           ← 文字数を数えます。(名前が何文字かを数えます。)
 50:E$(0)="O"              ← E$に "O" を代入
 60:FOR C=1 TO B           ┐ Oを左から右へ名前の文字数分動かします。
 70:PRINT E$(0)            │ 動くスピードは20ラインのWAIT文で指定され
 80:E$(0)="_"+E$(0)        │ ます。
 90:NEXT C                 ┘
100:WAIT : PRINT B$(0)     ← 停止時間の指定を解除し、名前を表示します。
```

注）LENは文字列の文字数を求める関数です。145ページを参照してください。

6−3）ポーズ（PAUSE）

PRINT命令が表示をおこなってプログラムの実行を停止、あるいはWAIT命令により停止時間が指定されるのに対して、このPAUSE命令（ポーズ）は停止時間が一定(約0.85秒)に決っており、その時間だけ表示をおこなって次のプログラムに実行が移るところが異っているだけで、そのほかはPRINT命令と同じです。

=プログラムの実行開始方法と、スタート状態の違いについて=

プログラムの実行開始方法には次の方法があります。

① RUN命令によるもの

RUN ENTER

RUN 式 ENTER

RUN "ラベル" ENTER

② GOTO命令によるもの

GOTO 式 ENTER

GOTO "ラベル" ENTER

- GOTO ENTER ではエラー（ERROR 1）になります。

③ 定義付けキーによるもの

DEF に続くA、S、D、F……

これらの開始方法により、変数のクリアや、状態の解除などに違いがありますので次に示します。

RUNによる実行	GOTOによる実行 定義付けキーによる実行
●表示フォーマット（USING）指定を解除	●表示フォーマット指定を保持
●ウエイト（WAIT）指定を解除	●ウエイト指定を保持
●配列変数をクリア※ （DIM指定を解除）	●配列変数を保持 （DIM指定を保持）
●FOR−NEXTスタック GOSUBスタックをクリア	●FOR−NEXTスタック GOSUBスタックをクリア
●READ文に対するDATAの初期化	●READ文に対するDATAの初期化はおこなわない。
●PRINT=LPRINT指定を解除 （PRINT=PRINTを指定）	●PRINT=LPRINTあるいはPRINT=PRINTの指定を保持

※RUN命令によりプログラムを実行しますと、それまでメモリー（プログラム・データエリア）上に確保されていた配列変数（DIM指定）、単純変数およびA（27）以上の変数は消去されます。

ただし、固定変数（A～ZあるいはA$～Z$）は保持されています。

7. そのほかのBASIC

7−1）READとDATA、RESTORE
（リードとデータ、リストア）

(1) READとDATA

変数にデータを入れる（代入する）方法として、LETやINPUTのほかにREAD命令とDATA命令の組み合わせがあります。
これはREADに続いて変数を書き、そのデータをDATAに続いて書きます。この場合の一般形は次のようになります。

```
READ　変数，変数，……
 ⋮
DATA　式，式，……　　（DATA　文字列，文字列，……）
```

```
 50：READ　A，B
  ⋮
110：DATA　3，4，5
120：DATA　60，"G="，"H="
  ⋮
200：READ　C，D，E$，F$
```

Aには3、Bには4が代入されます。

Cには5、Dには60、E$には"G="
F$には"H="が代入されます。

このようにREADの後の変数と、DATAの後のデータは1対1で対応します。したがって変数の形とデータの形が一致していなければなりません。すなわち、数値変数に対応するデータは数値でなければなりませんし、文字変数に対応するデータは文字列でなければなりません。
1つのプログラムの中に何回でもREAD文およびDATA文を書くことができますがデータは何ラインに分けて書いても一連のデータと見なされ、若いラインのデータから順番に変数に代入されます。
また、複数のプログラムが書き込まれ、それぞれのプログラムにDATA文がある場合でも、それは一連のデータと見なされます。
したがって、それぞれのプログラム内のデータを使用するときは、プログラムのスタートラインに、次で説明のRESTORE命令を書き込んでください。

（例）

```
10:READ A,B
20:G=√(A*A+B*B)
30:READ C,D,E$,F$
40:H=C* COS D
50:PRINT E$;G;F$;H
60:DATA 3,4,5
70:DATA 60,"G=","␣␣H="
80:END
```

Aには3、Bには4が代入されます。

Cには5、Dには60、E$には"G="、F$には"　H="が代入されます。

(2) RESTORE

READ命令の実行時、DATA命令で指定されているデータのどのデータを読み込むかは常に計算機に記憶されていますが、この読み込むデータの順番を強制的に変えるときに使用します。

RESTORE 式（あるいはRESTORE "ラベル"）

この形の場合、この命令を実行した次のREAD命令の変数には、RESTORE命令の式の値（あるいはラベル）で指定されるラインに書かれているDATA命令の最初のデータから1対1に対応して代入されます。

（例）

```
10  DATA 10, 20
20  DATA 30, 40
30  DATA 50, 60
 :
100  READ A, B, C, D
110  RESTORE 20
120  READ E, F, G, H
```

A、B、C、Dにそれぞれ、10、20、30、40が代入されます。

E、F、G、Hにはそれぞれ30、40、50、60が代入されます。

RESTORE

この形の場合、この命令を実行した次のREAD命令の変数には、プログラムの最も若いラインに書かれているDATA命令のデータが1対1に対応して代入されます。

7−2）ON　GOTO、ON　GOSUB

(1)　ON　GOTO命令（オン・ゴートゥ）

GOTO命令についてはすでに述べましたが、ON　GOTO命令はGOTO命令の機能を拡大したものです。ON　GOTO命令の一般形は次のようになります。

ON　式　GOTO式$_1$，式$_2$，式$_3$，……
　　（ON　式　GOTO "ラベル"，"ラベル"，…）

この命令ではONの次の式の値が１の場合はGOTOに続いて書かれている式$_1$で示されるラインへジャンプし、ONの次の式の値が２の場合は式$_2$で示されるラインへとジャンプします。もし、式の値が、１より小さいときやGOTOに続いて書かれている式の数より大きい値のときは、ON　GOTO命令の書かれている次のラインへ実行が移ります。

（例）

```
  :
50:ON A GOTO 100,110,120
  :
```

Aの値が１のときは100ラインへ、２のときは110ラインへ、３のときは120ラインへ、ジャンプします。Aの値が、１、２、３以外のときは、次のラインへ実行が移ります。

```
 10:DIM X$(0)*24
 20:INPUT "1,2,3,4_?_",A
 30:ON A GOTO 100,110,12
    0,130
 40:GOTO 20
 50:X$(0)=""
 60:FOR S=1 TO B
 70:X$(0)=X$(0)+Z$
 80:NEXT S
 90:PRINT X$(0): GOTO 20
100:B=2:Z$="A": GOTO 50
110:B=5:Z$="B": GOTO 50
120:B=11:Z$="C": GOTO 50
130:B=4:Z$="ABC_": GOTO
    50
```

← 30行：Aの値が１のときは100ラインへ、２のときは110ラインへ、３のときは120ラインへ、４のときは130ラインへジャンプします。Aの値が、１，２，３，４以外のときは、次のラインへ実行が移ります。

(2)　ON　GOSUB命令（オン・ゴーサブ）

ON　GOTO命令と同じ形で指定すれば、ONに続く式の値により、GOSUBに続いて書かれた式あるいはラベルで示すラインへサブルーチンジャンプします。

注）GOTOおよびGOSUBに続く式には、二次元配列変数（B（1，2）、B（5，3）など）を使用することはできません。

7−3）REM（リマーク）

この命令は、プログラムの実行には関係なく、プログラムリストなどをわかりやすくする目的でプログラムの先頭や途中に注釈を入れておくものです。

```
(例) 10  REM  KINRI KEISAN
     ⋮
     200  REM  SUBROUTINE
     ⋮
```

7−4）STOP（ストップ）

この命令はプログラムの一時停止命令です。この命令を実行しますと「BREAK IN 200」のように、実行したラインナンバーを持ったブレークメッセージを表示して実行を停止します。

```
(例)   ⋮
     60：STOP
       ⋮
```

STOP命令により実行が一時停止している状態（ブレーク状態）のときはマニュアル操作が可能になるため、変数の状態を調べたり、変数に数値を代入したりする場合によくこの命令が使用されます。

- STOP命令により停止した実行を再開させるときはCONT ENTER と操作してください。

7−5）CONT（コンティニュー）

CONT命令は、プログラムの実行が一時停止しているときに、プログラムの実行を再開させるための命令です。したがって、この命令はプログラムに書き込むことはできずRUNモードでのマニュアル操作でのみ有効となります。この命令の一般形は次のようになります。

CONT ENTER

- プログラムの一時停止状態（ブレーク状態）とは次の状態です。
  ①プログラム実行中にSTOP命令で一時停止した状態
  ②プログラム実行中に ON/BRK キーを押して一時停止した状態
  ③そのほか、PRINT命令の実行などにより実行が停止している状態

7-6) AREAD (オートリード)

この命令は定義付キーにより、プログラムの実行を開始したとき、表示されていた内容を指定された変数に読み込む命令で、定義付プログラム（121ページ参照）で、ラベルに続いて書きます。

この命令の一般形は次のようになります。

AREAD　変数

（例）　10 "A" : AREAD　A

　　　　⋮

　　　100 "B" : AREAD　A$

○ この命令は、プログラム実行開始直後以外はスキップ（無視）されます。

○ プログラムの実行を開始したとき、表示がプロンプト表示になっていたときは、指定されている変数はクリアされます。

○ プログラムの実行を開始したとき、PRINT命令による表示がなされていたときは、次の内容が読み込まれます。

```
たとえば次のようなプログラムを実行した場合
10 "A" : PRINT "ABC", "DEFG"
20 "S" : AREAD  A$ : PRINT  A$
RUNモード
DEF  A  → ABC     DEFG
DEF  S  → DEFG
```

● PRINT式，式　あるいはPRINT "文字"，"文字" により表示がなされていた場合は 右側に表示されていた内容が読み込まれます。

● PRINT式；式；式…により表示がなされていた場合は最初（いちばん左）に表示されていた内容が読み込まれます。

● PRINT "文字"；"文字"；"文字"…により表示がなされていた場合は最後に指定されている文字が読み込まれます。

8. 特別な機能

8－1）乱数関数（RND，RANDOM）

(1) RND命令(ランダム)

RND命令は乱数を発生させる機能を持つ関数です。

RND　X　においてXの値により次のような乱数を得ることができます。

○Xが1以上の場合

Xが整数のとき：　1からXの値以下の乱数を発生します。（1≦RNDX≦X）

Xが小数部を含むとき：　1からXの整数部に1を加えた値以下の乱数を発生します。（1≦RNDX≦(INT X)+1）ただし、この場合は小数部の値によって乱数の発生に差が出ます。

○Xが負数の場合：　直前に発生した乱数(あるいは乱数列)と同じ乱数を発生させるために、初期値を一定にします。

○Xが0から1未満の場合：1未満の乱数を発生します。（0＜RNDX＜1）

(例)

```
 10:DIM B(7)
 20:FOR A=0 TO 1
 30:D= RND -1
 40:FOR B=0 TO 3
 50:B(B+A*4)= RND 9
 60:NEXT B
 70:NEXT A
 80:USING "##"
 90:PRINT B(0);B(1);B(2)
    ;B(3);B(4);B(5);B(6)
    ;B(7)
100:END
```

30ラインの代入分は同じパターンの乱数を発生させるために入れています。このラインを削除すれば同じパターンの乱数は発生されなくなります。（変数Dは、Dにかぎらず、プログラム内で他に使用していない変数であればよい。）

表示例

```
 7 1 4 3 7 1 4 3
```

4個の乱数を2回表示させています。

○乱数の有効桁数は10桁です。

(2) RANDOM命令（ランダマイズ）

RND命令の使用に先立って乱数のタネを植えつけるものです。

RNDは電源オンからやり直すと、常に同じ乱数を発生します。しかし、電源オン後にRANDOM命令を実行すれば同じ乱数を発生することは、まず、なくなります。

（例）

```
10:RANDOM
 :
```

8−2）INKEY$（インキードル）

プログラムの中で、この命令が実行されたとき、いずれかのキーが押されていれば、その内容を読み込んで指定された変数に代入します。なお、このときのキー操作に対しての表示はおこなわれません。

INKEY$の一般形は次のようになります。

文字変数＝INKEY$

（例）

```
10:Z$= INKEY$
20:IF Z$="" THEN 10
30:PRINT "---";Z$;"---"
40:GOTO 10
```

（10〜20行：このラインをくり返し実行し、キーが押されるのを待ちます。）

〔プログラムの実行を終了するときは ON/BRK CA/C·CE と押してください。〕

◉ INKEY$を使用するときのご注意

① プログラムのはじめにINKEY$がありますと、プログラムをスタートさせたとき、スタートキーを読み取ってしまうことがありますので、注意が必要です。たとえば、上記のプログラムにおいて

```
10 "Z" :Z$=INKEY$
 :
```

として、DEF Z でプログラムをスタートさせたとき、Z キーが読み取られる場合があります。

② ENTER 、C·CE 、DEF 、SHIFT 、↓ 、↑ 、◄ 、► キーがINKEY$に読み取られた場合、その内容はNUL（空白）になります。

なお、ON/BRK キーはプログラムの一時停止キー（ブレークキー）として働きます。

③ INKEY$命令は SHIFT キーや DEF キーが押されていれば、そのキーを読み込みます。したがって、SHIFT キーに続いて押したときに働く機能や入力される記号 DEF キーに続いて押したときに働く機能などを読み込むことはできません。

8−3）音発生機能（BEEP）

本機には音発生機能がついていますので、プログラムの途中で音を出し、計算機の操作をより効果的に、また楽しくすることができます。

◉ BEEP命令（ビープ）

BEEP命令は音の発生（発生回数）を指定する命令です。BEEP命令の一般形は次のようになります。

BEEP 式

式の値で示す数だけピ音を発生します。

式の値は整数部のみ有効で、9.999999999E99以下の正の数値でなくてはなりません。

（例）

```
50 BEEP 5
```
←ピ音を5回発生

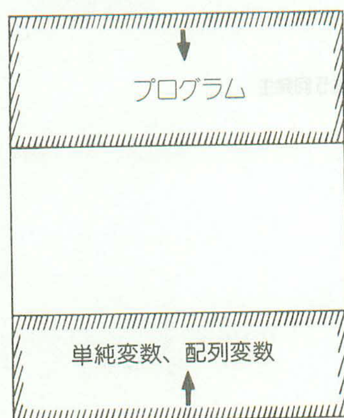
8−4）MEM（メム）

プログラム・データエリアの中で空いている部分（プログラム、配列変数などで使用されている部分以外）のバイト数を確認するための命令です。この命令の一般形は次のようになります。

MEM

MEM ENTER

A=MEM



MEM ENTER でこの部分のバイト数を表示します。

●プログラムの大きさ（バイト数）は、次の操作で求めることができます。

RUNモード

CLEAR ENTER （単純変数や配列変数などを消去します。）

3534−MEM ENTER →プログラムのバイト数を表示

PC−1402の場合は9678にしてください。

8−5）論理積、論理和、否定（AND, OR, NOT）

⑴ 論理積（AND：アンド）

論理積は次のような値を取る関数です。

```
1 AND 1  =1
1 AND 0  =0
0 AND 1  =0
0 AND 0  =0
```

たとえば41と27の論理積（AND）を計算する場合は、これらの数値を２進数に変換して、それぞれの桁のANDを取り、その結果を10進数に変換します。

```
41 AND 27 = 9

     ┌101001……41
AND ┤
     └011011……27
      ─────────
      001001…… 9
```

⑵ 論理和（OR：オア）

論理和は次のような値を取る関数です。

```
1 OR 1 =1
1 OR 0 =1
0 OR 1 =1
0 OR 0 =0
```

たとえば41と27の論理和（OR）はAND命令と同様、２進数に変換して、それぞれの桁のORを取りその結果を10進数に変換します。

```
41 OR 27=59

    ┌101001……41
OR ┤
    └011011……27
     ─────────
     111011……59
```

(3) 否定（NOT：ノット）

10進数を16桁の２進で表わすと次のようになります。

10進	16桁の２進
32767	0111111111111111
⋮	
2	0000000000000010
1	0000000000000001
0	0000000000000000
−1	1111111111111111
−2	1111111111111110
⋮	⋮
−32768	1000000000000000

ここで２進数0000000000000001の否定（NOT）を取りますと次のようになります。

NOT　　0000000000000001
（否定）　1111111111111110

このように、各桁の１を０に、０を１に反転することを“否定(NOT)を取る”といいます。さて、いま１とNOT　１を加え合わせると、どうなるでしょうか。

　　0000000000000001（１）
＋）1111111111111110（NOT　１）
　　1111111111111111（－１）

このようにすべての桁が１になります。これは上記の数値表で見ると、10進数では－１になります。

すなわち、１＋NOT　１＝－１となります。

このように数値Ｘとその否定（NOT　Ｘ）の間には

　　Ｘ＋NOT Ｘ＝－１

の関係があります。

このことからNOT Ｘ＝－Ｘ－１

すなわち　　NOT Ｘ＝－(Ｘ＋１)

の関係式が導き出されます。

この関係式から

　NOT　０＝－１
　NOT－１＝０
　NOT－２＝１

となることがわかります。

8－6）条件式の結合（AND, OR）

これまで、IF文の中で使用しました条件式は

A＞5
A＄＝"A"
A＊B＞＝C+20

などのような、2つの値の大小比較の式が1つのみでしたが、これらの大小比較の式を2つ以上結合して、1つの条件式をつくることができます。たとえば"Aは5よりも大きくかつ10よりも小さい"という条件を式で表わす場合、これまではIF文を2つ使って

```
10  IF  A<=5  THEN  30
20  IF  A<10  THEN  60
30
 :
```

のような形で表わす必要がありましたが、AND命令を使えば

```
10  IF  A>5  AND  A<10  THEN  60
```

のように1つの条件式にすることができます。

(1) AND命令（アンド）

2つ以上の条件のすべてを満足するような条件を1つの式で表わすときに用います。

（例）Aは0よりも大きくかつ6より小さい場合の条件式

A＞0　AND　A＜6

(2) OR命令（オア）

2つ以上の条件のいずれかを満足するような条件を1つの式で表わすときに用います。

（例）Aは0よりも小さいか、あるいは12より大きい場合の条件式

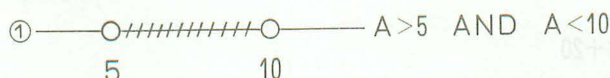
A＜0　OR　A＞12

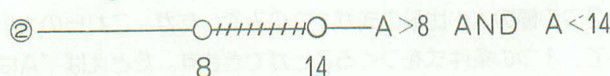
○AND、OR、を使った条件式の例

① A>5 かつ A<10

② A>8 かつ A<14

③ ①と②のAND







(A>5 AND A<10) AND (A>8 AND A<14)

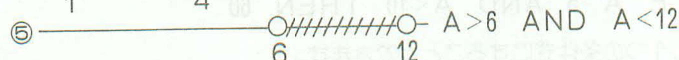
この場合カッコはなくても同じです。

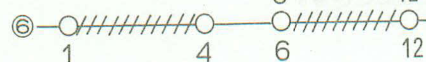
④ A>1 かつ A<4

⑤ A>6 かつ A<12

⑥ ④と⑤のOR







(A>1 AND A<4)OR (A>6 AND A<12)

=参　考=

論理計算は次のように構成することにより、論理和（OR)、論理積（AND）を形づくることができます。

① 論理和　（論理計算）＋（論理計算）

例　(A<0) + (A>8)　Aが0より小さいか、あるいは8より大きいときに1の値を取る

(B>0) + (C>0)　Bあるいは Cが0より大きいとき、およびB、Cともに0より大きいとき1（あるいは2）の値を取る

② 論理積　（論理計算）＊（論理計算）

例　(B>1) ＊ (B<6)　Bが1より大きく、かつ6より小さいとき1の値を取る

③ 特殊な使い方の例

例　B=(A=1) ＊3+ (A=2) ＊7　Aが1のときBは3になりAが2のときBは7になる。その他のときはBは0になる。

9. 文字の編集

この項では、文字を分断したり、結合したり、また、文字列の大小比較など、文字と文字の扱いについて説明します。

9－1）文字列の分断

(LEFT$, RIGHT$, MID$)

(1) LEFT$関数(レフトドル)

この関数は、ある文字列の左側から何文字かを取り出す関数です。この関数の一般形は次のようになります。

LEFT$(文字変数，式)

あるいは

LEFT$("文字",式)

(例)

```
10:A$="ABCDE"
20:B$= LEFT$ (A$,3)
30:PRINT B$
```

←A$の文字列の左側3文字、すなわちABCを取り出し、B$に代入する。

(2) RIGHT$(ライトドル)

この関数はLEFT$関数とは逆に、右側から何文字かを取り出す関数です。一般形は次のようになります。

RIGHT$(文字変数,式)

あるいは

RIGHT$("文字",式)

(例)

```
10:A$="ABCDE"
20:B$= RIGHT$ (A$,3)
30:PRINT B$
```

←A$の文字列の右側3文字、すなわちCDEを取り出し、B$に代入します。

⑶　MID$(ミッドドル)

この関数は文字列の中間の文字だけを取り出す関数です。この関数の一般形は次のようになります。

　　　　　　　　↱左側何文字目から取り出すか指定

MID$(文字変数，式$_1$，式$_2$)

　　　　　　　　　　　　↳何文字を取り出すか指定

MID$("文字"，式$_1$，式$_2$)

(例)

```
10:A$="ABCDE"
20:B$= MID$ (A$,2,3)
30:PRINT B$
```

← A$の文字列の左側2文字目から3文字、すなわちBCDを取り出し、B$に代入します。

9－2）文字列の結合

いくつかの文字列を結合させて、新しい文字列を作る場合は、"文字列の加算"をおこないます。たとえば、A$="ABC"、B$="DEF"、"XYZ"の3つの文字列を結合させる場合は

A$+B$+"XYZ"

のように加算の形で指定することができます。

注）通常、文字固定変数は7文字まで記憶することができ、文字単純変数は16文字まで記憶することができます。

また、文字配列変数は特に指定していない場合は16文字まで記憶することができます。

文字列を変数に代入する場合はこれらの記憶可能文字数を越えないようにしてください。

もし、越える可能性がある場合はDIM指定により文字配列変数の大きさを広げてお使いください。

9－3）文字数を数える方法(LEN)

１つの文字列の中に含まれる文字の数(記号、スペース、数字も含みます)は計算機により、たちどころに求めることができます。

LEN関数(レングス)

たとえば

A=LEN "ABC1256"

のように指定して実行すれば"ABC1256"の文字数７が変数Aに代入されます。
この関数の一般形は次のようになります。

LEN { "文字" / 文字変数 }

```
 10:DIM B$(0)*20,E$(0)*2
    0
 20:WAIT 30
 30:INPUT "NAMAE_WA_?_",        ←名前を入力します。
    B$(0)
 40:B= LEN B$(0)                ←文字数を数えます。
 50:E$(0)=""                    ←E$(0)の内容をクリアします。
 60:FOR J=1 TO B
 70:Z$= MID$ (B$(0),J,1)        ←文字を右から１字ずつ取り出します。
 80:E$(0)=E$(0)+Z$              ←取り出した文字をE$(0)に加えます。
 90:PRINT E$(0)                 ←E$(0)の内容を表示します。
100:NEXT J
110:GOTO 50                     ←50ラインにもどって、くり返し
```

●プログラムをスタートさせ、名前を入れれば、右側の文字から順番に表示していきます。

9－4）数値と文字の変換(VAL，STR$)

数値を文字列として扱いたい場合や、逆に文字列として指定されている数字を数値として扱いたい場合がありますが、このようなとき、STR$あるいはVAL関数を使います。

⑴ STR$関数(ストリングドル)

この関数は数値を文字列の形に変換する関数です。たとえば、B=1234のとき

A$=STR$ B

として実行すれば、A$には"1234"という文字列が代入されます。この関数の一般形は次のようになります。

STR$ 式

(例) A$=STR$120　　A$="120"と同じになります。

B$=STR$(1.2*3)　　B$="3.6"と同じになります。

⑵ VAL関数(バリュー)

VAL関数はSTR$関数とは逆に文字列を数値に変換する関数です。たとえば

B$="1234"のとき

A=VAL B$

として実行すればAには1234という数値が代入されます。この関数の一般形は次のようになります。

VAL {"文字" / 文字変数}

(例) A=VAL"120"　　Aに120が代入されます。

B=VAL"3.2*4="　　Bに3.2が代入されます。

VAL関数により数値に変換できる文字列は数値を表わすことのできる数字(0〜9.)、符号(+、-)、指数部を示す記号(E)で構成されている文字列で、他の文字や記号を含んでいないことが必要です。もし、1つの文字列の中に他の文字や記号が含まれている場合は、それから右の文字列は無視されます。

9－5）アスキーコードの変換(ASC，CHR$)

(1) アスキー(ASCII)コード

アルファベット、数字、記号などを計算機が記憶したり、処理したりする場合は、すべて計算機が取り扱いやすい数値に変換します。たとえばアルファベットのAは計算機内では65(10進数)という数値(コード)になっています。（実際には2進数の01000001となっています。）同様にBは66、Cは67というようにコードを決めています。このコードの決めかたに何種類かあって、その1つがアスキーコード（ASCII code）です。

(2) ASC関数(アスキー)

この関数は文字や記号、数字などをアスキーコードに変換する関数です。たとえば「Z」という文字のアスキーコードを知りたい場合は

　　A＝ASC "Z"

として実行すれば、Aには「Z」のアスキーコードが10進数の90として代入されます。この関数の一般形は次のようになります。

```
        { "文字"
    ASC {
        { 文字変数
```

なお、文字が2文字以上指定された場合は、頭の文字のみがアスキーコードに変換されます。

(3) CHR$関数(キャラクタドル)

この関数はASC命令とは逆の関数で、10進数のアスキコードを文字や記号（0～9の数字を含みます）に変換する関数です。たとえば、アスキーコード「90」の文字を知りたい場合は、

　　A$＝CHR$　90

として実行すれば、A$には"Z"が代入されます。CHR$の一般形は次のようになります。

　　CHR$　式

なお、本機では、文字や記号(これらを総称してキャラクタ：Characterといいます）に対応しているコードはØ及び32～95までと251、252です。ただしØはNUL(空白)です。(243ページ参照)

9－6）文字列の比較

2つの文字列を比較して、その大小判断をおこなうことができます。（アスキーコードの大きさによる大小判断）この判断を用いれば、たとえば文字列をアルファベット順に並べ替えることも容易になります。文字列の比較は、数値の比較に用いました等号、不等号、大小比較の記号を用いることによりおこないます。たとえば

"ABC"<"BC"
"AKIRA">"AKIO"
"YAMATO"="YAMATO"

（アスキーコードではAが65、Bが66、Cが67…となっています。したがって、文字列の比較では、AはBより小さく、BはCよりも小さくなります。また、数字はアルファベットよりも小さくなります。）

のように書き表わすことができます。いま、文字列をA$とB$とした場合、次の種類の比較ができます。

A$=B$	A$とB$は同じ文字列か？
A$<B$	A$はB$よりも小さい文字列か？
A$>B$	A$はB$よりも大きい文字列か？
A$<=B$	A$はB$よりも小さいか等しい文字列か？
A$>=B$	A$はB$よりも大きいか等しい文字列か？
A$<>B$	A$とB$は同じ文字列ではないか？

これらはIF文の中に使用して大小判断に用いることができます。

＝式の中での文字列の長さについて＝

文字列の結合（加算）や、大小比較などの式において、文字列の長さ（文字数）の合計は80文字以内でなければなりません。80文字を超えますと文字記憶エリア（計算や比較のため、文字を一時記憶しておくところ）の容量をオーバーしてエラーになります。（ERROR　5）

たとえば、次の式では文字列の長さの合計は9文字になります。

X=LEN("ABC"+LEFT$("DEFGHI",2))

（この場合、ABCとDEFGHIをすべて文字記憶エリアに入れてから、必要な処理をおこないます。）

10. パスワード(PASS)

作成したプログラムを他の人に知られたくないときや、プログラムを変更されたくないときに、ある言葉や記号をパスワードとし、パスワードが与えられないかぎり、計算機内のプログラムを呼び出すことができないようにすることができます。(プログラムの秘密化)

たとえば、プログラムを書き込んだあと、

PASS"YAMADA" ENTER

と操作すれば、計算機内のプログラムがすべて秘密化し、呼び出すことができなくなります。これを呼び出すためにはもう一度同じパスワードを宣言して、秘密化したプログラム（以下秘密プログラムと呼びます）を解除しなければならず、したがって、パスワードを知らなければプログラムを見ることも、変更することもできません。

◎ PASS(パス)

パスワードを設定あるいは解除する命令です。この命令の一般形は次のようになります。

PASS"文字列" ENTER

└ 7文字までの英文字、数字、記号

●パスワードが宣言されていない状態のとき、PASS命令を実行すれば、そのとき計算機内にあるすべてのプログラムに対してパスワードが設定され、秘密プログラムになります。

秘密プログラムに対してLIST命令、↑ や ↓ キーなど、プログラムの呼び出しにかかわる命令や機能およびラインの追加・削除などの機能は働きません。

また、テープレコーダや他の出力機器に出力することもできません。

●秘密プログラムを解除する場合はもう一度同じパスワードを宣言します。(パスワードが違っているときはエラー(ERROR 9)になります。)

●計算機内にプログラムが入っていないときにPASS命令を実行するとエラー(ERROR 1)になり、パスワードは設定されません。

●秘密プログラムでないプログラムを秘密プログラムとしてテープに記録することができます。

秘密プログラムの記録、読み込みについては、157ページをご覧ください。

11. デバッグ

プログラムを実行した場合に、何かの誤りで迷走したり、思わぬ結果が出てきたりすることがあります。
このような場合、プログラムのリストを調べて、誤りを探しますが、それだけではなかなか見つけにくい場合があります。
このようなとき、プログラムを1ラインずつ実行させ、その経過をたどりながらプログラムの誤りを探していけば、見つけやすくなります。
ここでは、1ラインずつ経過をたどりながらプログラムを実行する方法について説明します。(このような方法を**トレース**といいます。また、プログラムの誤りを探し、修正することを**デバッグ**(虫取り)といいます。)

(1) デバッグのしかた

① トレースによるデバッグはRUNモードでおこないますので、RUNモードにしてください。

② TR ON ENTER と押し、トレースモードを指定します。

③ RUN ENTER と押してプログラムの実行を開始します。(最初のラインの実行が終われば実行は停止します。)

④ その後は ↓ キーを押します。(1ラインだけ実行して停止します。)ただし、INPUT命令によるデータの入力や、PRINT命令により停止しているときの再開は、通常のプログラムの実行と同じように ENTER キーでおこないます。

⑤ プログラムの実行順序の確認、各ライン実行後の変数の内容確認などをおこないながらトレースを進め、プログラムが正しく実行されているかどうかチェックします。
もし、正しく実行されないときは、その原因を探して、修正します。

⑥ デバッグが終了すれば、TR OFF ENTER と押してトレースモードを解除してください。

次にごく簡単な例を示します。

〈プログラム例〉

```
10  INPUT "A=";A, "B=";B
20  C=A*2
30  D=B*3
40  PRINT  "C=";C;"␣D=";D
50  END
```

〈実行〉

RUNモード

TR ON [ENTER]		＞	
RUN [ENTER]		A=_	INPUT命令実行
8 [ENTER]	(データ入力)	B=_	
9 [ENTER]	(データ入力)	10：	←10ライン終了
[↓]		20：	←20ライン終了
[↓]		30：	←30ライン終了
[↓]		C=16. D=27.	←PRINT命令実行
[ENTER]		40：	←40ライン終了
[↓]		＞	←実行終了（プロンプト表示）

デバッグ動作でラインナンバーを表示しているときは、変数の内容をマニュアル操作によって呼び出し、所望の値になっているかどうかチェックできます。また、このとき [↑] を押せば、押している間止まっているラインの内容を表示します。（なお、[↑] キーを押して離したときは、プロンプト記号(＞)が表示されます。）

注）トレースモードは、TR OFF [ENTER] と押すか、[SHIFT] [CA C・CE] と押す、あるいはスイッチをOFF側にして電源を切るまで指定された状態が保たれます。

(2) プログラムの途中で実行を停止させてチェックする場合

プログラムの途中にSTOP命令を書いておけば、プログラム実行途中でSTOP命令を実行し、ブレークメッセージを表示して実行が停止します。

このとき

① マニュアル操作で変数の内容をチェックする。

② トレースモードを指定して [↓] キーの操作にて以降のラインを1ラインずつデバッグ操作する。

③ トレースモードを解除して CONT [ENTER] と操作し、通常の実行状態にて続行する。

● 通常のプログラム実行中に [ON BRK] キーを押しますと、現在実行しているラインの終わりで実行を停止し、ブレークメッセージを表示します。このときも前記同様3つの操作をおこなうのが普通です。なお、ブレーク状態（停止状態）のときに [↑] を押せば、押している間停止しているプログラムラインが表示されます。

● トレースモードを指定しない場合でも、[ON BRK] キーやSTOP命令などにより一時停止しているプログラムを [↓] キーにより1ラインずつ実行させていくことができます。この場合も表示はトレースモードが指定されている場合と同じようになります。

なお、[↓] キーを押したままにしますと、普通のプログラム実行時と同じように、プログラムが実行されます。

12. プリンタに関する命令

これ以降に説明しますプリンタに関する命令は別売のCE-126Pが接続されているときに有効となる命令です。
CE-126Pが接続されていない場合、これらの命令は無視されます。
また、CE-126Pの電源スイッチ（POWER）がOFF位置にあるときにも、これらの命令は無視されます。

12－1）プリント命令（LPRINT）

計算機の表示命令として、PRINT命令がありましたが、これに代わるプリンタの印字命令としてLPRINT（ラインプリント）命令があります。
このLPRINT命令の一般形はPRINT命令とほとんど同じ形で指定することができます。

① LPRINT { 式 / "文字" / 文字変数 }

この形で指定した場合、数値はペーパーの右側に詰めてプリントされ、文字列は左端からプリントされます。
なお、文字列が24桁を越える場合は自動的に改行してプリントします。（CE-126Pの1行の印字桁数は24桁です。）

② LPRINT { 式 / "文字" / 文字変数 } , { 式 / "文字" / 文字変数 }

この形で指定した場合は、1行の印字桁数24桁を左右12桁に分け、，の左右の内容をプリントします。このときも、12桁の範囲内で数値は右ヅメ、文字は左ヅメにします。
なお、表示内容が12桁を越える場合、数値は仮数部の下位桁を切り捨てて12桁以内でプリントし、文字は頭から12桁をプリントします。

③ LPRINT { 式 / "文字" / 文字変数 } ; { 式 / "文字" / 文字変数 } ;…; { 式 / "文字" / 文字変数 }

この形で指定した場合、ペーパーの左端から連続的にプリントされます。プリント内容が24桁を越える場合は自動的に改行され、最大79字までプリントされます。なお、79字目が数値の途中になる場合はエラーになります。

12−2）PRINT→LPRINT指定

本機はPRINT命令を必要に応じて表示命令と印字命令とに切替えて用いることができます。

たとえば、計算機本体のみで使用しているときは、PRINT命令を表示命令として表示部に計算結果を表示させ、CE-126Pと接続しているときは印字命令として計算結果などをプリントさせることができます。

⊙指定・解除

CE-126Pが接続されているとき、マニュアルあるいはプログラムで次の命令が実行されれば、PRINT命令はすべてLPRINT命令と同様に働きます。

PRINT=LPRINT

この命令は

PRINT=PRINT

命令で解除することができるほか、RUN命令の実行、SHIFT CA C·CE の操作、電源スイッチで電源をオフ・オンすることにより解除され、PRINT命令は通常の表示命令にもどります。

なお、RUN命令の実行でPRINT=LPRINTの指定は解除されますので、マニュアルでPRINT=LPRINT命令を実行させ、かつ有効に働かせるには、次のような方法を用いてください。

- 命令実行後、GOTO命令でプログラムをスタートさせる。(129ページ参照)
- 命令実行後、定義付けキーによりスタートさせる。(121ページ参照)
- PRINT命令などにより、プログラムがストップしているときに、この命令を実行する。など

＝ご注意＝

数値を出力する場合、表示部に表示する場合と、プリンタで印字する場合は、数値を表わす方式がかわる場合があります。
たとえば、次のプログラムを実行した場合

```
10:A=1234567,B=-1234567
   8
20:PRINT A,B
30:LPRINT A,B
```

20ラインのＰＲＩＮＴ命令では

```
1.2E 06-1.2E 07
```

と表示されますが、30ラインのＬＰＲＩＮＴ命令では

```
1234567.  -12345678.
```

と印字されます。
これは、表示が16桁であるのに対して、印字は１行が24桁になっているために、表示では固定小数点方式で表わせない場合でも、印字では固定小数点方式で表わせるからです。
なお、ＰＲＩＮＴ命令でも、ＰＲＩＮＴ＝ＬＰＲＩＮＴの指定がなされている場合は、ＬＰＲＩＮＴ命令と同じに動きます。

12－３）マニュアル計算でのプリント

マニュアル計算の結果は通常表示部に表示されるのみですが、ＣＥ-126Ｐが接続されているときは SHIFT ENTER(P↔NP) と操作して図に示します位置に ▬ マークを表示させておけば、計算の過程や結果がプリントされます。（プリントモード）



プリントをさせたくないときは、もう一度 SHIFT ENTER(P↔NP) と操作して ▬ マークを消してください。（ノンプリントモード）

- マニュアル計算においてエラーになった場合は、入力されたキー操作のみをプリントし、結果はプリントしません。そして、表示部にエラーの表示をおこないます。
- 入力した内容がＢＡＳＩＣ命令から始まっている場合は、プリントはおこなわれません。
- マニュアル計算において、ダイレクトアンサー機能によって計算した内容はプリントされません。
- また、電卓モードでの計算もプリントされません。

12－4）リストのしかた（ＬＬＩＳＴ）

ＬＬＩＳＴ（ラインリスト）は計算機に記憶されているプログラムを書き出す命令で、一般形は次のようになります。（ＰＲＯあるいはＲＵＮモードでのマニュアル操作により実行します。）

① ＬＬＩＳＴ

プログラムの先頭ラインから、すべてのプログラムを書き出します。

② ＬＬＩＳＴ 式

式の値で示されるラインを書き出します。

③ ＬＬＩＳＴ 式$_1$，式$_2$

式$_1$ の値で示されるラインから式$_2$ の値で示されるラインまでのプログラムを書き出します。

なお、式$_1$および式$_2$の値がピッタリ一致するラインがない場合は、それぞれの式の値よりも大きくかつ最も近いラインナンバーが指定されます。

ただし、式$_1$と式$_2$で指定されるラインが同じになる場合はエラー（ＥＲＲＯＲ　１）になります。

④ ＬＬＩＳＴ 式$_1$，

式$_1$ の値で示されるラインから、最後のラインまでのプログラムを書き出します。

⑤ ＬＬＩＳＴ，式$_2$

プログラムの先頭ラインから、式$_2$ の値で示されるラインまでのプログラムを書き出します。

＝プリンタに関するご注意＝

- プリンタの動作中に紙づまりなどが発生しますと計算機にエラー（ＥＲＲＯＲ　８）が検出されます。
  このときはプリンタにつまった紙を取り除いてから C·CE キーでエラーを解除してください。
- 強度の外来ノイズをうけた場合などに、プリンタが不要なプリントを続け、止まらなくなることがありますが、この場合は ON/BRK キーで動作を止めた後、 C·CE キーを押してください。
- ＣＥ-126Ｐの電池電圧が低下しますと、プリンタは動作しなくなります。
  この場合は電池を新しいものに交換してください。
- ＣＥ-126Ｐの紙送りキー（ ⇡ ）はプリントをおこなっているときは動作しません。

13. テープレコーダーに関する命令

ここではテープレコーダーを使用して、テープにプログラムやデータを記録したり、計算機に読み込んだり（転送したり）、あるいは記録されている内容を照合したりする命令について説明します。

13−1）プログラムの記録命令(CSAVE)

(1) CSAVE命令（カセット・セイブ）

プログラムをテープに記録する(セイブする)命令で、次のような形で指定します。

① CSAVE "ファイル名" ENTER
あるいは　CSAVE ENTER

○プログラムの記録

プログラムをテープに記録する場合はプログラムモードか実行モードでCSAVE命令を実行します。

（例）PROモードかRUNモードを指定
CSAVE "PROG. −1" ENTER

ファイル名として「PROG. −1」をテープに記録したのち、計算機内のプログラムをすべて記録します。

注）同じテープの同じ面（A面あるいはB面）に、同じファイル名で内容の違うプログラムを記録することは避けてください。同じファイル名のものが2つ以上ありますと、テープから読み込む（転送する）ときにまちがった内容を読み込む恐れがあります。

記録をおこなうとき、テープレコーダーのテープカウンターの数字を紙などに控えておけば、転送などをおこなうとき、記録した部分を簡単に探すことができます。

◉ ファイル名

ファイル名はプログラムなどにつける名前（見出し）のことで、ファイル名をつけてプログラムなどを記録しておけば、あとで転送などをおこなうとき、計算機によりファイル名を探し出して転送などをおこなうことができますので便利です。

なお、ファイル名を省略することもできます。

CSAVE命令は、計算機内のプログラムが秘密プログラムのときは働きません。したがって秘密化されているプログラムをテープに記録するときはPASS命令で解除してください。

なお、テープに記録するプログラムを秘密化するときは次の方法でおこなってください。

(2) 記録するプログラムの秘密化

テープに記録するプログラムを秘密化するときは次の形で記録してください。

② CSAVE "ファイル名", "パスワード" ENTER
あるいはCSAVE, "パスワード" ENTER

この形でCSAVE命令を実行すれば、プログラムはパスワードつきのプログラムとして記録され、後述のCLOAD命令などで計算機に読み込まれたときは、秘密プログラムとなります。

したがって、解除するにはPASS命令でパスワードを宣言しなければなりません。

●パスワードは最大7文字までの英文字、数字、記号などを用いることができます。

＝ご注意＝

プログラムをテープに記録したあと、必らず159ページの命令および175ページの方法で照合をおこなってください。

もし、照合を何回かおこなって、その都度エラーになる場合は次の計算をおこなって結果を確認してください。

PC−1401の場合

RUNモード

CLEAR ENTER

(3534−MEM)/120 ENTER　および

(3535−MEM)/120 ENTER

この計算のどちらかの結果が整数値（1、2、3……29）になったときは、プログラムの実行に支障のない範囲で、1ラインの削除（マルチステートメントにする）あるいは追加をおこなってから、もう一度記録し直してください。

(例)

```
10:A=0, B=0            ⇨    10:A=0, B=0
 ∫                            ∫
200:NEXT I                   200:NEXT I:END
210:END
```

そして照合をおこなってください。

それでも、なおエラーになるときは、テープの記録する位置を変えるか、別のテープをお使いください。

また、テープレコーダーが原因の場合も考えられますので、CE−124やCE−126Pの取扱説明書を参考にテープレコーダーが適当かどうかお調べください。

13−2）プログラムの転送命令（CLOAD）

◉ CLOAD命令(カセット・ロード)

テープに記録されているプログラムを計算機に転送するための命令で、次のような形で指定します。なお、この命令はマニュアル操作でのみ実行できます。

① CLOAD "ファイル名" ENTER
あるいは　CLOAD ENTER

○プログラムの転送

テープに記録されているプログラムを計算機に転送する場合はプログラムモードか実行モードでCLOAD命令を実行します。

（例）PROモードかRUNモードを指定

CLOAD "PROG. −1" ENTER

ファイル名が「PROG. −1」のプログラムを検索して、これを計算機に転送します。

ファイル名をつけずにCLOAD　ENTER　と操作した場合はテープレコーダーが動き始めて最初に記録されているプログラムを計算機に転送します。

●転送がおこなわれているときは、表示部の右端の桁に ＊ マークが表示されます。そして、転送が終れば、＊ マークは消えます。
ファイル名を検索しているときは、まだ転送がおこなわれていませんので ＊ マークは表示されません。
これは、のちほど説明しますCLOAD?およびINPUT#命令でも同じです。

注1）指定されたファイル名が検索できなかったときは、テープが回り終っても計算機はファイル名を検索し続けます。この場合は ON/BRK キーを押して検索を止めてください。これは、のちほど説明しますCLOAD?、およびINPUT#命令でも同じです。

注2）CLOAD命令を実行中にエラーが発生しますと、計算機内のプログラムは無効になります。

13−3）照合命令（CLOAD?）

◉ CLOAD?命令（カセット・ロード?）

計算機に記憶されているプログラムと、テープに記録されている内容との照合をおこなう命令で、次のような形で指定します。なお、この命令はマニュアル操作による実行のみ可能です。

① CLOAD? "ファイル名" ENTER

あるいは　　CLOAD? ENTER

この命令により、もし不一致が生じたときは、エラー（ERROR　8）になります。

不一致がなければプロンプト表示になります。

○プログラムの照合

計算機に記憶されているプログラムとテープに記録されているプログラムを照合する場合はプログラムモードあるいは実行モードでCLOAD?命令を実行します。

（例）PROモードかRUNモードを指定

CLOAD? "PRO−1" ENTER　　ファイル名が「PRO−1」のプログラムを検索し、その内容と、計算機のプログラムとを照合します。

●プログラムの照合がおこなわれているときは表示部の右端の桁に ＊ マークが表示されます。

照合が終れば ＊ マークは消えます。

13－4）データの記録命令（PRINT＃）

◎PRINT＃命令（プリント・クロスハッチ）

計算機内に記憶されているデータをテープに記録する命令で、次のような形で指定します。（マニュアル操作、プログラムとも実行できる命令です。）

PRINT＃ "ファイル名" ；変数名，変数名…
あるいは　PRINT＃　変数名，変数名…

● 変数名

変数名は次の形で指定します。

① 固定変数……A、B、X、A(26)など
　　　　　　　C＊、A(10)＊など
② 単純変数……AA、B2、XY$など
③ 配列変数……B(＊)、CD(＊)、N$(＊)など

① 固定変数の記録

固定変数は、変数名を個個に指定すれば、指定した変数の内容が記録されます。

PRINT＃"DATA1"；A，B，X，Y

ファイル名「DATA1」をつけて、変数A、B、X、Yの内容をテープに記録します。

また、指定した固定変数から後の固定変数をすべて記録したいときは、変数名の後に＊記号をつけて指定します。

PRINT＃ "D－2" ；D＊

ファイル名「D－2」をつけて、固定変数D～Zおよび固定変数の延長としてA(27)以降の変数が確保されていれば、その変数すべての内容をテープに記録します。

PRINT＃　E，X$，A(30)＊

ファイル名なしで、固定変数EとX$の内容および固定変数の延長として確保されているA(30)以降の変数すべての内容をテープに記録します。

注1）A(1)～A(26)の指定はA～Z（あるいはA$～Z$）の指定と同じです。ただし、DIM命令により配列Aが定義されている場合は、固定変数をA(　)の形で指定することはできません。

② 単純変数（2文字変数）の記録

単純変数は、変数名を個個に指定して、その変数の内容を記録します。

```
PRINT# "DM-1" ;AB, Y1, XY$
```

ファイル名「DM−1」をつけて、変数AB、Y1、XY$の内容をテープに記録します。

③ 配列変数の記録

配列変数は、配列名（*）の形で指定することにより、その配列要素のすべての内容を記録します。

```
PRINT# "DS-2" ;X (*), Y$ (*)
```

ファイル名「DS−2」をつけて、配列Xの要素（X(0)、X(1)…）と配列Y$の要素（Y$(0)、Y$(1)…）の内容をすべてテープに記録します。

注）配列の1要素だけを記録することはできません。
なお、固定変数および固定変数の延長として使用されるA(　)の形の変数は1要素だけを記録することができますが、DIMにより定義された配列Aはほかの配列と同様の指定方法でのみ記録することができます。

● 変数名を指定せずにPRINT#命令を実行しますとエラー（ERROR　1）になります。

＝ご注意＝

固定変数の延長のA(27)以降の変数や単純変数、および配列変数はPRINT#命令を実行する前にプログラム・データエリアに確保されていなければなりません。
もし、確保されていない変数を記録しようとしますとエラーになります。

13−5）データの転送命令（INPUT＃）

⊙INPUT＃命令（インプット・クロスハッチ）

テープに記録されているデータを指定された変数に転送する命令で、次のような形で指定します。（マニュアル操作、プログラムとも実行できる命令です。）

INPUT＃〝ファイル名″；変数名，変数名，変数名…

あるいは　INPUT＃　変数名，変数名…

● 変数名

変数名は次の形で指定します。

① 固定変数……A、B、C、A(7)など
D＊、A(20)＊など

② 単純変数……AA、B3、CP$など

③ 配列変数……S(＊)、HP(＊)、K$(＊)など

① 固定変数への転送

テープに記録されているデータを固定変数に転送する場合は、転送したい変数名を指定します。

INPUT＃〝DATA1″；A，B，X，Y

ファイル名「DATA1」がついてテープに記録されているデータを、変数A、B、X、Yに順番に転送します。

また、テープに記録されているデータを有るだけ、または入るだけ固定変数および固定変数の延長のA(　)変数に転送する場合は、始まりの変数名に＊記号をつけて指定します。

INPUT＃〝D−2″；D＊

ファイル名「D−2」がついてテープに記録されているデータを変数D～ZおよびA(27)以降に転送していきます。

INPUT＃　A(10)＊

テープが回りはじめて最初に記録されているデータファイルのデータを変数A(10)から始まって、添字がそれより大きい変数に転送していきます。
（J～Z、A(27)～に転送していきます。）

注1）配列Aが定義されているときは、A(　)の形で固定変数などの指定はできません。
また、A(27)以降の変数へのデータの転送はおこなわれません。

注2）固定変数およびその延長のA(　)の形の変数へのデータ転送は、記録されているデータファイルのデータがなくなれば終了します。

② 単純変数への転送

変数名を指定すれば、その変数にテープのデータが転送されます。

```
INPUT# "DM-1" ; AB, Y1, XY$
```

ファイル名「DM−1」がついてテープに記録されているデータをAB、Y1、XY$に転送していきます。

注1）テープに記録されたデータを単純変数に転送する場合、データと変数は、その形（数値変数か文字変数かの形）は必らず一致させてください。

注2）単純変数は、INPUT#命令を実行する前に、プログラム・データエリアに確保されていなければなりません。
もし確保されていなければエラーになります。
したがって、このようなときは事前に代入文で変数を確保してください。

```
AA=Ø [ENTER]
B1$= "A" [ENTER]
INPUT#  AA, B1$ [ENTER]
```

適当な数値や文字などを代入することにより変数を確保します。

③ 配列変数への転送

配列変数は、配列名（＊）の形で指定することにより、その配列要素にテープのデータが転送されます。

```
  ⋮
50  DIM  B(5)
60  INPUT#  "DS-4" ; B(*)
```

ファイル名「DS−4」がついてテープに記録されているデータを配列Bの各要素（B(Ø)〜B(5)）に転送していきます。

注1）テープに記録されたデータを配列変数に転送する場合、データと変数は、その形（数値変数か文字変数かの形）、大きさ、長さが一致していなければなりません。
もし、これらが一致していなければエラーになります。

注2）配列変数は、INPUT#命令を実行する前に、プログラム・データエリアに確保されていなければなりません。
もし確保されていなければエラーになります。
したがって、事前にDIM命令により配列の定義をしてください。

＝ご注意＝

ＩＮＰＵＴ＃命令で指定した変数の数とテープに記録されているデータの数が違っている場合は次のようになります。

- 指定した変数の数よりテープに記録されているデータの数が多い場合は、指定した変数にデータを転送し、残ったデータは無視します。
- 指定した変数の数よりテープに記録されているデータの数が少ない場合は、変数に順番にデータがあるだけ転送され、残った変数はそれまで記憶していた内容が保持されます。ただし、このとき計算機はデータが転送されてくるのを待っている状態になっていますので、このときは ON/BRK キーで動作を止める必要があります。
- 変数名を指定せずにＩＮＰＵＴ＃命令を実行しますとエラー（ＥＲＲＯＲ　１）になります。

14. BASICモードでのキーの機能

次に主なキーの機能を説明します。

キー	機能
ON BRK	○オートパワーオフ機能により切れている電源を入れるときに押します。 ○プログラム実行中、プログラムを一時停止状態（BREAK状態……ブレーク）にします。 ○CLOAD命令などの実行中は、その実行を中止します
SHIFT	○π、へ、？のように、薄茶色でパネル上に記されている命令(機能)を指定します。 （例） SHIFT U(?) → ？が入力されます。 SHIFT A(INPUT) → INPUT が入力されます。
C·CE	○入力中の内容や表示をクリアするときに押します。(エラー状態を解除します。)
SHIFT CA	○計算機の状態を解除します。(クリアオール) ●プログラムの実行が一時停止状態にあるとき、実行を中止させます。 ●WAIT命令による時間指定を解除します。 ●表示フォーマット指定(USING指定)を解除します。 ●トレースモードの解除(TROFF状態にする) ●PRINT=LPRINT指定の解除 ●エラー状態の解除 その他
0 ～ 9	○数値を指定します。
·	○小数点を指定します。 ○命令入力の際の省略形の指定をおこないます。 ○USING命令内において、数値データの表示に小数点以下の表示をおこなう指定をします。
EXP または E	○指数部入力を指定します。(E キーはアルファベットキーです。)
/	○除算命令を指定します。
*	○乗算命令を指定します。 ○DIM命令やPRINT#、INPUT#命令の中で配列変数などを指定するときに使用します。

キー	機能
+	○数値の符号（正符号）を指定します。（一般に省略可） ○加算命令を指定します。
−	○数値の符号（負符号）を指定します。 ○減算命令を指定します。
=	○代入文において、右辺の内容を左辺で指定される変数に代入することを示します。 ○ＩＦ文の条件式などに使用します。
SHIFT ^	○べき乗命令を指定します。 ○ＵＳＩＮＧ命令内において、数値データの表示形式を浮動小数点方式（指数方式）に指定します。
SHIFT < SHIFT >	○ＩＦ文の条件式などに使用します。 ○論理式を指定します。
DEF	○Ａ、Ｓ、Ｄ、…などのラベルで定義されたプログラムの実行を開始するとき、そのラベルに対するキーを押す前に押します。 (Definableキー指定)
A 〜 Z	○アルファベットキーで、各命令や、変数名などを入力します。 ○Ａ、Ｓ、Ｄ…などは定義付けキーとして使用できます。
SPC	○プログラム入力あるいは文字列入力時にスペースを設けるためのキーです。 〝　〟内及びＲＥＭ文内以外のスペースはプログラム、演算などの実行時にはないものとみなされます。 ○定義付けキーとして使用できます。
,	○式と式の区切りや変数の区切り、コメントの区切りなどを指定します。 ○定義付けキーとして使用できます。
SHIFT INPUT SHIFT IF 〜 SHIFT CLOAD	○それぞれのキーに定義されているＢＡＳＩＣ命令を入力します。

キー	機　　能
SHIFT ！ SHIFT " SHIFT ＃ SHIFT ＄ SHIFT ％ SHIFT ＆ SHIFT ？ SHIFT ： SHIFT ； SHIFT ＠	○それぞれのキャラクターシンボルを入力します。 "…文字列の指定をおこないます。 ラベルの指定をおこないます。 ＃…ＵＳＩＮＧ命令内において、数値データの表示フォーマットを定義します。 ＰＲＩＮＴ＃命令およびＩＮＰＵＴ＃命令 を指定するときに使用します。 ＄…文字変数を指定します。 ＆…ＵＳＩＮＧ命令内において、文字列の表示フォーマットを定義します。 16進数値を示します。 ？…ＣＬＯＡＤ？命令を指定するときに使用します。 ：…１ラインに複数のステートメントを定義するときの区切りを指定します。 ；…式や変数の区切り、コメントの区切り、命令文と変数の区切りなどを指定します。 ＠！％…" "の中で文字列として使用します。
（	○カッコの指定をおこないます。
）	
►	○カーソルの右シフトをおこないます。 ○プレイバック命令の実行をおこないます。 ○プログラム内容が表示されているとき、カーソルが表示されていない場合にカーソルを呼び出します。 ○マニュアル操作時のエラーを解除します。
◄	○カーソルの左シフトをおこないます。 ○そのほか ► キーと同じ。
SHIFT INS	○カーソルが示す位置に、１ステップ容量の挿入スペース（ただし□表示）を設けます。
SHIFT DEL	○カーソルが示す位置の内容を削除します。
CAL	○電卓モード(ＣＡＬモード)を設定します。
BASIC	○ＢＡＳＩＣモードを設定します。 ○ＢＡＳＩＣモードにおいて、実行モード（ＲＵＮモード）とプログラムモード（ＰＲＯモード）を切換えて指定します。

キー	機能
sin 〜 x^2 SHIFT sin^{-1} 〜 SHIFT n!	○それぞれのキーに定義されている関数を入力します。 ○数値のみが入力された直後および計算結果表示中はそれぞれの関数計算を実行します。 (ただし、y^x、$\sqrt[x]{y}$ 、→xy、→$r\theta$、πの機能は除きます)
hyp	○双曲線関数を入力するときに使用します。
SHIFT arc hyp	○逆双曲線関数を入力するときに使用します。
ENTER	○プログラムのラインの終了命令を指定します。 ○プログラムの書き込みをおこないます。 ○マニュアル計算の実行あるいはBASIC命令のマニュアル操作による実行を命令します。 ○INPUT命令やPRINT命令で、一時停止しているプログラムの再スタートを命令します。
SHIFT P↔NP	○別売のプリンタが接続されているとき、マニュアル計算のプリント↔ノンプリントモードを指定します。プリントモードでは表示部下の PRINT 位置に ▬ マークが表示され、計算の過程や結果がプリントされます。

[↓] [↑] キーについては、モードの指定および計算機の状態により、次のように働きます。

モード	状態	[↓]	[↑]
RUN	プログラム実行中		
	プログラムの一時停止中 INPUT命令実行時 PRINT命令実行時 ブレーク状態	次のラインを実行（1ラインずつ実行して停止します。）	押している間、実行しているラインあるいは実行したラインを表示
	プログラム実行時のエラー状態		押している間、エラー発生のラインを表示
	トレースモードオン状態	デバック動作を実行	押している間、実行しているラインあるいは実行したラインを表示
	そのほかの状態	直前に実行された計算の最終結果を表示（ラストアンサー機能）	同　左
PRO（モードをPROモードに切換え、プログラムラインが示されていないとき）			
	プログラムの一時停止中	実行を停止しているラインを表示	同　左
	エラー発生後	エラー発生のプログラムラインを表示	同　左
	その他	先頭ラインを表示	末尾ラインを表示
PRO（プログラムラインが表示されているとき）		次のプログラムラインを表示	1ライン前のラインを表示

● [ENTER] キーは表示上ではスペースと同じです。

● キー入力待ちの状態で、約10分間キー入力がなければ自動的に電源が切れます。（Auto power off：オートパワーオフ機能）

この場合は [ON BRK] キーを押せば、電源が入ります。ただし、このときモードは電卓モードになります。

注）[F↔E]、[SHIFT] [TAB]、[⇕]、[SHIFT] [STAT]、[SHIFT] [Δ%]、[x↔M]、[RM]、[M+]、[+/−] および統計計算関係のキーはBASICモード（RUNモードとPROモード）では働きません。

◎表示シンボル

表示部の上側に表示されるシンボルの意味を示します。

BUSY　：プログラムを実行していることを示します。

DEF　：［DEF］キーが押されたことを示します。このシンボルが表示されているとき、アルファベットキーなどが定義付けキーとして働きます。

DEG　：ディグリー〔°〕
RAD　：ラディアン〔rad〕
GRAD　：グラード〔g〕

三角関数や逆三角関数の計算において、取り扱う角度の単位を示すシンボルです。

$(90° = \frac{\pi}{2}〔rad〕= 100^{g})$

SHIFT　：［SHIFT］キーが押されたことを示します。このシンボルが表示されているときは、キーの上側のパネルに薄茶色で記されている機能が指定されています。

HYP　：［hyp］キーが押されたことを示します。このシンボルが表示されているときは、双曲線関数の入力ができます。

第５章

その他

■ もくじ ■

（ページ）

１．オプション機器のご案内…………172

２．計算機とテープレコーダーの操作方法…………173

2−1)ＣＥ−126Ｐを使用される場合…………173

2−2)ＣＥ−124を使用される場合…………177

３．ＰＣ−1245用、ＰＣ−1250／ＰＣ−1251用プログラムを使用する場合の注意点…………183

3−1）ＰＣ−1245用プログラムの修正・変更…………182

3−2）ＰＣ−1250／ＰＣ−1251用プログラムの修正・変更…………185

3−3）そのほかの注意点…………186

４．電源について…………187

５．仕様…………190

1．オプション機器のご案内

PC−1401／PC−1402には、次の別売周辺機器が使用できます。

●CE−124

カセットインターフェイス

CE−152などのテープレコーダーとPC−1401／PC−1402をつなぐためのインターフェイスです。

●CE−126P

サーマルプリンタ（リモート機能内蔵カセットインターフェイス付）

サーマルペーパー（感熱紙）にプログラムをリストしたり、計算結果や処理結果を書き出したりすることのできるプリンタです。

また、カセットインターフェイスを内蔵していますので、CE−152などのテープレコーダーと接続して、プログラムやデータをテープに記録したり、その内容をテープから読み込んだりすることができます。

●CE−152

ポケットコンピュータ専用カセットテープレコーダー

CE−124あるいはCE−126Pを介して、プログラムやデータをテープに記録、転送ができます。

2. 計算機とテープレコーダーの操作方法

2－1）CE－126Pを使用される場合

プログラムなどをテープに記録する、あるいはテープから転送する場合の計算機とテープレコーダーの操作方法について説明します。

注）テープレコーダーは別売のポケットコンピュータ専用カセットテープレコーダーCE－152を使用されているものとして説明いたします。

(1) 準　備

◉ 計算機とCE－126Pを接続し、テープレコーダ（CE－152）と接続してください。
（接続方法はCE－126P及びCE－152の取扱説明書を参照してください。）

◉ テープレコーダーに関する注意

- 転送あるいは照合をおこなうときは、記録をおこなったテープレコーダーを使用してください。
  記録したテープレコーダーと転送あるいは照合するテープレコーダーが異なる場合は、転送や照合ができないことがあります。
- テープレコーダーのヘッドなどが汚れていますと、ひずみの増加、レベルの低下などがおこりますので、ヘッドの清掃をおこなってください。
- テープは一般に市販のものをご使用ください。ただし、極端に周波数特性の悪いもの、また、傷や折目があるものは使用しないでください。
- 一度使用したテープに新しくプログラムなどを書き込む場合は、書き始める前を少し消去してから記録命令を実行してください。

(2) テープへの記録方法

例としてプログラムを記録する方法について説明します。（マニュアル操作の場合）
〈プログラム〉

```
10 INPUT "HANKEI␣R=";A
20 S=π*A*A
30 PRINT "MENSEKI␣S=";S
40 END
```

手順	操　　　作	備　　　　考
(1)	CE-126Pのリモートスイッチを OFF側にしてください。	
(2)	テープをテープレコーダーにセットしてください。	テープの中の記録されていない部分をさがし、記録する位置をきめてください。
(3)	記録する内容を計算機に入れてください。 たとえば、前ページのプログラムを計算機に書き込んでください。	
(4)	CE-126PのリモートスイッチをON側にしてください。	
(5)	テープレコーダーの記録ボタン(●)を押し込んで、録音状態にしてください。	
(6)	記録命令を実行します。 プログラム：CSAVE"ファイル名" ENTER データ：PRINT#"ファイル名"；変数名，変数名… ENTER (例) CSAVE "PR−1" ENTER	記録命令が実行されますと、まず最初に約8秒間無信号区間が記録（ピー音が記録）され、そのあとにファイル名とプログラムなどが記録されます。(プログラムなどの記録中はピピピピ……あるいはビビビビ…というような音が出ます。)
(7)	記録が終了しますと、プロンプト表示になり、テープが止まります。 (巻きもどし、早送りなどをおこなうときは、CE-126PのリモートスイッチをOFF側にして、おこなってください)	●記録が終りましたら、次ページの照合をおこなってください。

●プログラムの実行において、データの記録をおこなう場合(PRINT#命令)には、プログラム実行前にテープレコーダーの操作をしておいてください。

記録をおこなうとき、記録を始めたときのテープカウンタの数値を紙などに控えておけば、記録位置や、記録されていない位置を探し出すときに便利です。

(3) 計算機とテープの内容の照合

計算機に記憶されているプログラムと、テープに記録されているプログラムとの照合を取る場合(マニュアル操作でのみ実行可能)には次のようにおこないます。

手順	操作	備考
(1)	CE-126Pのリモートスイッチを OFF側にしてください。	
(2)	照合する内容が記録されているテープをテープレコーダーにセットしてください。	テープカウンターなどを見ながら照合する内容の記録されている位置の近く(少し前)までテープを送ってください。
(3)	音量調整つまみ、音質調整つまみを矢印方向いっぱいに回しておいてください H◀TONE▶L　VOLUME◀MAX	
(4)	CE-126PのリモートスイッチをON側にしてください。	
(5)	テープレコーダーの再生ボタン(▶)を押し込んで再生状態にしてください。	
(6)	照合命令を実行します。 CLOAD?"ファイル名" ENTER (例) CLOAD?"PR-1" ENTER 照合がおこなわれているときは表示部の右端の桁に＊マークが表示されます。	ファイル名は記録したファイル名を入れてください。ファイル名を省略した場合はテープが回り出して最初の記録内容と、計算機内の内容との照合をおこないます。 ファイル名にまちがいがないか、よく確認しながら操作してください。
	両方の内容が一致している場合は実行を終了しプロンプト表示になります。 もし、エラー表示(ERROR 8)になった場合は、もう一度最初から照合をおこなってください。 それでもなお、エラーになる場合は、157ページの確認をおこなって、もう一度"記録"からおこなってください。	

＝ご注意＝

テープレコーダーの再生ボタン(▶)が押されていて、転送命令や照合命令が実行されているときに、早送りボタン(▶▶)や巻きもどしボタン(◀◀)を押さないでください。
早送りや巻きもどしをおこないますと、転送や照合が正常におこなわれません。

(4) テープからの転送

テープに記録されているプログラムあるいはデータを計算機に転送する（テープより読み出す）場合は次のようにおこなってください。（マニュアル操作の場合）

手順	操　作	備　考
(1)	CE-126PのリモートスイッチをOFF側にしてください。	
(2)	転送する内容が記録されているテープをテープレコーダーにセットしてください。	テープカウンターなどを見ながら転送する内容が記録されている位置の近く（少し前）までテープを送ってください。
(3)	音量調整つまみ、音質調整つまみを図の矢印方向いっぱいに回しておいてください。 H◀TONE▶L　VOLUME◀MAX	
(4)	CE-126PのリモートスイッチをON側にしてください。	
(5)	テープレコーダーの再生ボタン（▶）を押し込んで再生状態にしてください。	
(6)	転送命令を実行します。 プログラム：CLOAD“ファイル名” ENTER データ：INPUT#“ファイル名”；変数名，変数名，… ENTER （例） CLOAD “PR−1” ENTER	ファイル名は記録をしたファイル名を入れてください。 ファイル名を省略した場合はテープが回り出して最初に記録されている内容が転送されます。 転送がおこなわれているときは表示部の右端の桁に＊マークが表示されます。 転送が終了すればプロンプト表示になります。
(7)	もし、途中でエラー（ERROR　8）になった場合は、もう一度最初から転送をおこなってください。 それでもなお、エラーになる場合は、もう一度“記録”からおこなってください。	ファイル名にまちがいがないか、よく確認しながら操作してください。

●プログラムの実行において、転送をおこなう場合（INPUT#命令）には、プログラム実行前にテープレコーダーの操作をしておいてください。

2－2）CE－124を使用される場合

注）テープレコーダーは別売のポケットコンピュータ専用カセットテープレコーダーCE－152を使用されているものとして説明いたします。

(1) 準 備

◉ 計算機、カセットインターフェイス（CE－124)、テープレコーダー（CE－152）を接続してください。（接続方法はCE－124及びCE－152の取扱説明書を参照ください。）

◉ テープレコーダーに関する注意

- 転送あるいは照合をおこなうときは、記録をおこなったテープレコーダーを使用してください。
  記録したテープレコーダーと転送あるいは照合するテープレコーダーが異なる場合は、転送や照合ができないことがあります。
- テープレコーダーのヘッドなどが汚れていますと、ひずみの増加、レベルの低下などがおこりますので、ヘッドの清掃をおこなってください。
- テープは一般に市販のものをご使用ください。ただし、極端に周波数特性の悪いもの、また、傷や折目があるものは使用しないでください。
- 一度使用したテープに新しくプログラムなどを書き込む場合は、書き始める前を少し消去してから記録命令を実行してください。

(2) テープへの記録方法

例としてプログラムを記録する方法について説明します。（マニュアル操作の場合）

〈プログラム〉

```
10 INPUT "HANKEI␣R=" ; A
20 S=π*A*A
30 PRINT "MENSEKI␣S=" ; S
40 END
```

手順	操　作	備　考
(1)	テープをテープレコーダーにセットしてください。	テープの中の記録されていない部分をさがし、記録する位置をきめてください。 （テープの始めの無録音部分はさけてください。）
(2)	記録する内容を計算機に入れてください。	たとえば、前ページのプログラムを計算機に書き込んでください。
(3)	テープレコーダーの一時停止ボタン(Ⅱ)を押し込んで、一時停止状態にしたうえで、記録ボタン(●)を押して、記録可能な状態にしてください。	
(4)	記録命令を入力してください。 プログラム：CSAVE"ファイル名" データ：PRINT#"ファイル名"; 変数名，変数名…… （例）CSAVE "PR−1"	
(5)	記録命令を実行します。 一時停止ボタン(Ⅱ)を押して一時停止状態を解除し、ENTER キーを押してください。	記録命令が実行されますと、まず最初に約8秒間無信号区間が記録（ピー音が記録）され、そのあとにファイル名とプログラムなどが記録されます。(プログラムなどの記録中はピピピピ…あるいはビビビビ…というような音が出ます。)
(6)	記録が終了しますと、プロンプト表示になります。 テープを巻きもどして、"照合" をおこなってください。	"照合" については180ページを参照してください。

●プログラムの実行において、データの記録（PRINT#命令の実行）をおこなう場合は、PRINT#命令の前にSTOP命令を入れ、プログラムの実行を停止させるようにしてください。

（例）

```
       :
150：STOP：PRINT# "ABC"；F，B(*)
       :
```

そして、STOP命令によりプログラムが停止しているときに、テープレコーダーの操作をおこなってください。

次に、CONTとキー入力してからテープをスタートさせ、テープが回りはじめてから ENTER キーを押せばプログラムが再スタートして、STOP命令に続くPRINT#命令が実行されて、データの記録がおこなわれます。

●プログラムを停止させる方法としては、PRINT命令を用いる方法もあります。

（例）

```
       :
140：PRINT "TAPE  START"
150：PRINT# "ABC"；F，B(*)
       :
```

ただしこの場合、プログラムの中にWAIT命令が使用されているときは、その指定を解除するなどの配慮が必要になります。

記録をおこなうとき、記録を始めたときのテープカウンタの数値を紙などに控えておけば、記録位置や、記録されていない位置を探し出すときに便利です。

(3) 計算機とテープの内容の照合

計算機に記憶されているプログラムと、テープに記録されているプログラムとの照合を取る場合(マニュアル操作でのみ実行可能)には次のようにおこないます。

手順	操作	備考
(1)	照合する内容が記録されているテープをテープレコーダーにセットしてください。	テープカウンターなどを見ながら照合する内容の記録されている位置の近く(少し前)までテープを送ってください。
(2)	音量調整つまみ、音質調整つまみを矢印方向いっぱいに回しておいてください。 H◀TONE▶L　VOLUME◀MAX	
(3)	テープレコーダーの一時停止ボタン(II)を押し込んで、テープレコーダーを一時停止の状態にしてください。 続いて再生ボタン(▶)を押して、再生状態にしてください。	
(4)	照合命令を実行します。 CLOAD? "ファイル名" ENTER (例) CLOAD? "PR-1" ENTER	ファイル名は記録をしたファイル名を入れてください。ファイル名を省略した場合はテープが回り出して最初の記録内容と、計算機内の内容との照合をおこないます。
(5)	テープレコーダーの一時停止ボタン(II)により一時停止状態を解除してください。	(4)の操作後、ただちに一時停止状態を解除してください。 照合がおこなわれているときは、表示部の右端桁に＊マークが表示されます。
	両方の内容が一致している場合は実行を終了しプロンプト表示になります。 もし、エラー表示(ERROR　8)になった場合は、もう一度最初から照合をおこなってください。 それでもなお、エラーになる場合は157ページの確認をおこなって、もう一度"記録"からおこなってください。	

(4) テープからの転送

テープに記録されているプログラムあるいはデータを計算機に転送する（テープから読み込む）場合は次のようにおこなってください。(マニュアル操作の場合)

手順	操　　作	備　　考
(1)	転送する内容が記録されているテープをテープレコーダーにセットしてください。	テープカウンターなどを見ながら転送する内容の記録されている位置の近く（少し前）までテープを送ってください。
(2)	音量調整つまみ、音質調整つまみを矢印方向いっぱいに回しておいてください。 H◀TONE▶L　VOLUME◀MAX	
(3)	テープレコーダーの一時停止ボタン(II)を押し込んで、テープレコーダーを一時停止の状態にしてください。 続いて再生ボタン(▶)を押して、再生状態にしてください。	
(4)	転送命令を実行します。 プログラム： CLOAD"ファイル" ENTER データ： INPUT#"ファイル名" ENTER (例) CLOAD"PR−1" ENTER	ファイル名は記録したファイル名を入れてください。
(5)	テープレコーダーの一時停止ボタン(II)により一時停止状態を解除してください。	(4)の操作後、ただちに一時停止状態を解除してください。

手順	操　　作	備　　考
	転送が終了しますと、プロンプト表示になります。 もし、途中でエラー(ERROR 8)になった場合は、もう一度最初から転送をおこなってください。 それでもなお、エラーになる場合は、もう一度"記録"からおこなってください。	転送がおこなわれているときは、表示部の右端の桁に＊マークが表示されます。

●プログラムの実行において、転送をおこなう場合（INPUT＃命令）には、プログラムにおいてINPUT＃命令の直前にSTOP命令を入れてプログラムの実行を停止させ、テープレコーダーの操作をおこなってください。

3. PC−1245用　PC−1250／PC−1251用　プログラムを使用する場合の注意点

3−1）PC−1245用プログラムの修正・変更

本機はPC−1245とほぼ同一のBASIC命令を持っていますが、PC−1245に比べ、何点か機能が異っています。

従ってPC−1245用プログラムを本機で使用する場合は、以下に示します内容について修正が必要です。

(1)　＊命令を省略した乗算の修正

PC−1245では、A＊BやC＊DをABやCDのように＊命令を省略することができましたが、本機ではABやCDは単純変数と見なされます。

したがって＊命令を省略した乗算には＊命令を入れてください。

（例）A=SIN BC → A=SIN(B＊C)

(2) DIM命令によるA（ ）変数の定義の変更

PC−1245では、たとえばDIMA(30)を実行した場合、固定変数の延長として、A(27)〜A(30)が確保されていましたが、本機では固定変数とは別の配列Aとして、A(Ø)〜A(30)が確保されます。

したがってこの場合は次のように変更してください。

DIMA(30)　→　A(30)=Ø

(3) テープへのデータ入出力命令の変更

PC−1245では、たとえばPRINT#Cと指定しますと、C以降の固定変数の内容をすべて記録しますが、本機では変数Cの内容のみを記録します。

したがって、変数の指定方法を変更してください。

(例) PRINT#A　→　PRINT#A*

INPUT#C　→　INPUT#C*

(4) FOR−NEXT ループ終了後のループ変数値の違い

本機とPC-1245ではFOR−NEXTループ実行終了後のループ変数値が異なります。したがって、ループ終了後のループ変数値を判断条件などに使用している場合は修正してください。

(例)
```
10 FOR I=Ø TO 10
      ∫
50 NEXT I
60 IF I=10 THEN 100
```

このようなプログラムでは60ラインの条件式を

```
60 IF I=11 THEN 100
```

のように変更します。

（PC−1245に比べ、1回のきざみ値分、ループ変数の値が多くなります。ただし、ループの実行回数は同じです。）

(5) [=] キーへの定義付け変更

本機では [=] キーは定義付けキーとして働きません。
したがって、[=] キーに定義付けしているプログラムは、ほかのキーに定義付けしてください。

（例）

```
        :                          :
100 "=":………     →     100 "N":……
        :                          :
```

(6) 指数シンボル" E "の変更

本機では指数シンボルをアルファベットと同じ" E "を使用していますので、修正してください。

```
A=1.234E5     →     A=1.234 E 5
B=E6          →     B=1 E 6
```

(7) キャラクタ・コードの変更

PC−1245とPC−1401／PC−1402ではキャラクタ・コードが一部異っています。したがってCHR$命令により次のコードを指定している場合は、指定するコードを変えてください。

キャラクタコード	PC−1245のキャラクタ	PC−1401／PC−1402のキャラクタ
39（&27）	ニ	'
91（&5B）	√	[
93（&5D）	π	]
96（&60）	E	なし（エラー）
251（&FB）	なし（エラー）	π
252（&FC）	なし（エラー）	√

注）上に示しますように、PC−1401／PC−1402はキャラクタ ニ と E は持っていません。また、PC−1245はキャラクタ ' [ ] を持っていません。

以上の修正・変更が必要です。
ただし、PC−1245用プログラムをテープから読み込んだ場合、(6)の指数シンボルの変更は計算機内で自動的におこなわれます。

3－2）PC－1250／PC－1251用プログラムの修正・変更

本機でPC－1250／PC－1251用プログラムを使用する場合は、PC－1245のプログラムを使用するときの修正・変更に加えて、次のような修正・変更が必要です。

PC－1250／PC－1251は表示桁が24桁ですが、本機は16桁です。

したがいまして、PRINT、PAUSE、INPUT命令など、表示に関連した部分でエラーが発生したり、一部表示できない部分が出るなど、不都合な場合が出てきます。したがって、この場合は次のような変更をおこなってください。

(1) PRINT式　でUSING指定が総桁で16桁を越えているときはERROR　7になります。（ただし、整数部の指定桁が11桁を越えている場合は、越えた部分が無視されます。）

17桁以上

```
10 USING "########.########"
20 PRINT A
```

●この場合はUSING指定を16桁以内にしてください。

(2) PRINT式，式　でUSING指定の整数部が8桁を越えている場合（符号、小数点を含む）はERROR　7になります。

9桁以上

```
10 USING "#########.##"
20 PRINT A, B
```

●この場合はUSING指定の整数部を8桁以内にしてください。

(3) INPUT "……"；A　の形でメッセージ(入力ガイダンス)が15桁を越える場合、越えた桁数だけメッセージの先頭部分が表示されません。

●この場合、メッセージを短かくしてください。

(4) そのほか、次のような場合は越えた内容が表示されません。

PRINT ｛式／"文字"／文字変数｝；｛式／"文字"／文字変数｝；……

この形で表示内容が16桁を越える。

PRINT ｛"文字"／文字変数｝，｛"文字"／文字変数｝

この形で、左あるいは右の文字列が8桁を越える

INPUT "……"，A

この形でメッセージが16桁を越える。

●この場合は、表示内容が越えないように編集し直してください。

3−3）そのほかの注意点

⑴　本機はラインナンバーを1〜65279まで使用できるように拡張しています。
このため、ＰＣ−1245系（ＰＣ−1245、ＰＣ−1250、ＰＣ−1251）ではラインナンバーの記憶にメモリー２バイト分を使用するのに対して、本機では３バイト分を使用します。
ＰＣ−1245系のプログラムをテープから読み込んだときは、この変更が自動的におこなわれますが、ＰＣ−1251用プログラムが長い場合は、本機に入りきらなくなったり、プログラム実行時、変数が確保できなくなって容量オーバー（ＥＲＲＯＲ６）になる場合があります。

⑵　ＰＣ−1245系のプログラムをテープから読み込んだとき、テープが止まっても、あるいは読み込み音（ピー音）が止まっても１〜２秒計算機が読み込み状態（ＢＵＳＹシンボル、＊マーク表示）になっています。
これは、⑴で説明のラインナンバーのバイト数の変更をおこなっているためです。

⑶　本機はＰＣ−1245系が持っているＣＨＡＩＮ命令を持っていません。
したがって、ＣＨＡＩＮ命令が用いられているＰＣ−1245系のプログラムをテープから読み込んだ場合、表示ではＣＨＡＩＮ命令は“〜”シンボルで示され、プログラム実行時は無視されます。

⑷　ＰＣ−1245系の計算機において、ＭＥＲＧＥ命令（本機は持っていません）により複数のプログラムを読み込んで、一度に記録したテープから、本機にプログラムを転送しようとしたときは、先頭（１番目）のプログラムのみが読み込まれます。

⑸　**本機はＰＣ−１２４５系の計算機で作成され、テープに記録されたプログラムを読み込むことができますが、本機で作成し、テープに記録したプログラムをＰＣ−１２４５系の計算機に読み込ませることはできません。**

⑹　ＰＣ−1250、ＰＣ−1251、ＰＣ−1255用プログラムで、ＰＯＫＥおよびＣＡＬＬ命令が使用されている場合は、このプログラムを本機で実行させることはできません。本機で実行させますと、すべてのキーが働かなくなるなどの状態が発生する場合があります。

4．電源について

本機は本体のみで使用するときは、リチウム電池のみで動作しますが、CE-126Pが接続されている場合、CE-126P側の電圧が十分にあり、リチウム電池の電圧がさがっているときはCE-126P側の電源からも電気が供給されます。したがって、リチウム電池の消耗は少なくなります。

電池の交換時期

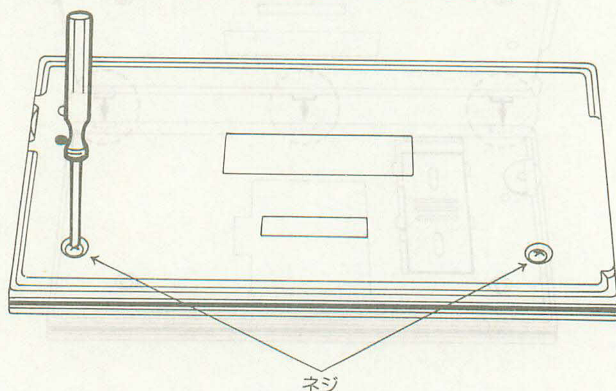
計算機右側面にあるコントラストボリュームを濃くなるほうにーぱいに回しても、正面から見た表示が薄く、見えにくくなった場合は電池の消耗を示していますので、速やかに新しい電池と交換してください。

（別売の周辺機器CE-126PやCE-124およびテープレコーダー（CE－152など）をお持ちの場合は、事前にプログラムやデータをテープに記録してください。）

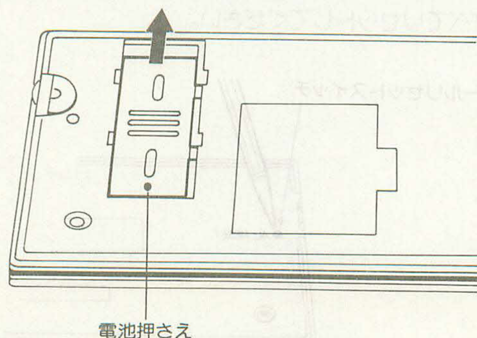
電池を入れる場合、および交換する場合は次の手順にしたがっておこなってください。

(1)　電源スイッチをＯＦＦ位置にしてください。

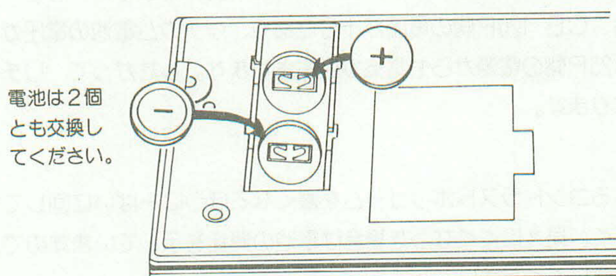
(2)　小型のネジ回しで裏ぶたの２本のネジをはずしてください。



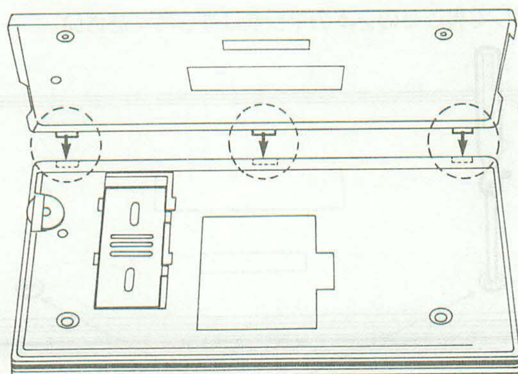
(3)　電池押さえを図の矢印の方向に動かして取りはずしてください。



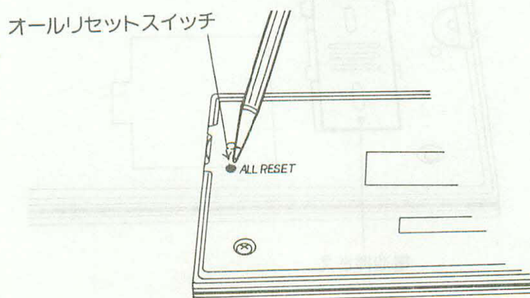
(4) 使用ずみの電池を取りはずし、新しいリチウム電池を2個入れます。
乾いた布で電池をよくふいて、プラス・マイナスをまちがえないように入れてください。



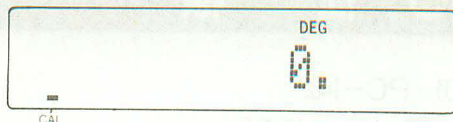
(5) 電池押さえを(3)と逆に取りつけてください。
(6) 裏ぶたを取りつけます。
裏ぶたのツメを本体のミゾに掛けてから裏ぶたを本体に合わせ、軽く押さえながらネジで留めてください。



(7) 電源スイッチをON側にして、計算機裏面のオールリセットスイッチを押し、計算機内の状態をすべてリセットしてください。



そしてそのあと、次のような表示になることを確認してください。



電池交換は前記の手順でおこなっていただくか、あるいはお買上げの販売店もしくは別紙一覧表に記載のサービス会社にお申しつけください。有料にて交換いたします。

◆電池は次の種類の新しい電池を2個ご使用ください。
　リチウム電池　CR－2032

〈電池使用上のお願い〉

1．電池は必ず同時に2個とも交換してください。
2．新しい電池と一度使用した電池をまぜて使用しないでください。
3．電池は必ず同種のものをご使用ください。
4．電池の＋、－の向きを本機の表示通り正しく入れてください。

〈ご注意〉

- **電池は幼児の手がとどかないところに置いてください。**
  万一飲み込んだ場合にはただちに医師にご相談ください。
- 計算機に内蔵された最初の電池は工場出荷時に組み込んでいますので約300時間に満たないうちに電池がなくなることがあります。
- 消耗した電池をそのままにしておきますと、液もれにより計算機をいためることがあります。
- 電池を火中に投入しないでください。火中に投入すると破裂するおそれがあります。

5. 仕　様

形　　名	PC−1401／PC−1402
計算桁数	10桁（仮数部）＋2桁（指数部）
計算方式	数式通り（優先順位判別機能つき）
プログラム言語	BASIC（ベーシック）
CPU	CMOS　8ビットCPU
システムROM	40キロバイト（40KB）
メモリー容量	システムエリア　約0.5キロバイト データ専用エリア　208バイト プログラム・データエリア　3534バイト（PC−1401） 9678バイト（PC−1402）
スタック	サブルーチン用　10段 FOR−NEXT用　5段 ファンクション用　16段 データ用　8段
基本計算機能	基本計算：加減乗除算 関数計算：三角関数、逆三角関数、双曲線関数、逆双曲関数、対数、指数、角度変換、べき乗、べき乗根、座標変換、開平計算、整数化、絶対値、符号関数、円周率、その他
編集機能	カーソル左右シフト（▶、◀） 挿　入（INS） 削　除（DEL） ラインアップ・ダウン（↓、↑）
メモリー保護	バッテリーバックアップ （プログラム、データなど電源オフ時内容保護）
表　　示	5×7ドットマトリックス　16桁液晶表示
使用温度	0℃〜40℃
電　　源	6V⎓（DC）：リチウム電池〈CR−2032〉2個
電池使用時間	約300時間（使用温度20℃で**55555555.**を連続的に表示させている場合。電池の種類や使用方法などにより多少の変動があります。） ● 1日当り1時間使用した場合は約4ヵ月使用できます。 ただし、1時間のうち、演算またはプログラム実行を10分間、表示状態を50分間おこなった場合。
消費電力	0.03W
外形寸法	幅170㎜×奥行72㎜×厚さ9.5㎜

重　　　量　150ｇ（ハードカバー、電池を含む）
付　属　品　ハードカバー、テンプレート１枚、リチウム電池２個（計算機に内蔵）、取扱説明書、ネームラベル、シャープサービス・お客様ご相談窓口一覧表

アフターサービスについて

◆修理を依頼されるときは

修理は、お買いあげの販売店に、本機の品名(計算機)、形名(PC−1401/PC−1402)および故障状態をくわしくご説明いただいて、お申しつけください。

ご転居・ご贈答品などでお買いあげの販売店に修理を依頼することができない場合は、もよりのシャープお客様ご相談窓口にお問い合わせください。

◆製品の保証について

① この計算機は取扱説明書の巻末に保証書がついています。保証書は販売店にて所定事項を記入いたしますので、内容をよくお読みのうえ大切に保存してください。

② 保証期間は保証書に記載されている期間です。
保証書の記載内容によりお買いあげの販売店または当社サービス会社が修理いたします。くわしくは保証書をご覧ください。

③ 保証期間中の修理などアフターサービスについておわかりにならない場合は、お買いあげの販売店、またはもよりのシャープお客様ご相談窓口にお問い合わせください。

④ 保証期間経過後の修理については販売店にご相談ください。修理によって機能が維持できる場合はお客様のご要望により有料修理いたします。

◆お問い合わせは

この製品についてのご意見、ご質問はもよりのお客様ご相談窓口（技術センター）へお申しつけください。付属の「シャープサービス・お客様ご相談窓口一覧表」のとおり、全国にお客様ご相談窓口（技術センター）を設けております。

▶ポケットコンピュータ友の会について◀

ポケットコンピュータ友の会は、ユーザーの皆様の自作ソフトの交流の場を提供します。

どうぞふるってご参加ください。

ご入会手続

下記に紹介致しておりますBASICソフトウェア集の巻頭にある、「ポケットコンピュータ友の会のご案内」での要領に従い皆様の作成されました興味あるオリジナルプログラム一編以上をお送りください。

会員特典

- 会員証を発行致します。(有効期間1年)
- 送付していただいたプログラムは、選考・編集の上、製本化し、出版社を通じて出版・販売いたします。なお、採用プログラム分につきましては記念品を贈呈させて頂きます。
- 上記の本を作る際の選考該当期間中にオリジナルプログラムをお送りいただいた会員の皆様にはこの本を一冊贈呈いたします。

送付先　〒639-11　奈良県大和郡山市美濃庄町492
シャープ株式会社　産業機器事業本部
OA事業推進本部　ポケットコンピュータグループ

BASICソフトウェア集のご案内

- ポケットコンピュータ　BASICソフトウェア集No.3　(PC−1401用)
- ポケットコンピュータ　BASICソフトウェア集No.4
(PC−1245/PC−1251
PC−1255/PC−1401用)

プログラム事例集

このページまで、各種機能の説明を読み進まれ、この機械がどのようなものかご理解いただけたと思います。ＢＡＳＩＣ言語で自由自在にプログラムが作成できるようになるには、この取扱説明書の理解に加えて、数多くのプログラム作りにチャレンジすることが大切です。電卓モードの使用についてもいろいろさわってみて、慣れるとよいでしょう。

以降のページには、ＢＡＳＩＣプログラムの例、電卓モードの使用例などをいくつか載せていますので参考にしてください。

（なお、当社はこの事例集の使用に際して生ずる金銭上の損害および逸失利益については、一切の責任は負いませんのであらかじめご了承ください。また、プログラムの使用に際し、精度不足や利用者の意図を十分達成できないこともありえますので、データチェックなど、十分確認をおこない、目的に合致しない場合は目的に合うよう修正の上、ご利用ください。）

注）●フローチャートは、プログラム構成の複雑なものに付けました。

目次

(タイトル)　(ページ)

●二重回帰分析……197
●二元配置法……202
(C)●平均値の比較……207
●直列・並列回路のインピーダンス……210
(C)●パルス回路……215
●方程式の根（ニュートン法）……216
●数値積分(シンプソン法)……220
●軟着陸……227
●タイプ練習……230
●集計表……233
(C)●割賦計算……239

注）(C)の付いたものは電卓モードの使用例です。

PROGRAM TITLE	二重回帰分析

計量値（x_i, y_i, z_i） $i=1$, …, nに何らかの関係があるとき、これらのデータを $z=ax+by+c$ で回帰します。

さらに、回帰式に従って推定します。身長（X），体重（Y）と胸囲（Z）の関係などに利用できます。

■使い方

1） DEF A …データ入力

ディスプレイ出力またはプリンタ出力の選択後、X，Y，Zにデータを入力します。終了時は"X＝"の表示の時に ENTER のみ押します。

2） DEF B ……分析結果の出力

ディスプレイ出力またはプリンタ出力の選択をします。

このあと、平均、分散、相関係数、及び回帰係数を出力します。

3） DEF C ……回帰式による推定

ディスプレイ出力またはプリンタ出力の選択後、X，Yのデータを入力すると、DEF B で決めた回帰係数によりZを推定します。

注）ディスプレイ出力またはプリンタ出力の選択では、"P＝P"はディスプレイ出力の意味で、"P＝LP"はプリンタ出力の意味です。

■内容

次の式により計算します。

n＝データ数

$S_{xx}=\sum(x_i-\bar{x})^2$

$S_{yy}=\sum(y_i-\bar{y})^2$

$S_{zz}=\sum(z_i-\bar{z})^2$

$S_{xy}=\sum(x_i-\bar{x})(y_i-\bar{y})$

$S_{yz}=\sum(y_i-\bar{y})(z_i-\bar{z})$

$S_{xz}=\sum(x_i-\bar{x})(z_i-\bar{z})$

回帰係数

$$a=\frac{Sxz\cdot Syy-Syz\cdot Sxy}{Sxx\cdot Syy-(Sxy)^2}$$

$$b=\frac{Syz\cdot Sxx-Sxz\cdot Sxy}{Sxx\cdot Syy-(Sxy)^2}$$

$c=\bar{z}-a\bar{x}-b\bar{y}$

推定は

$z=ax+by+c$の式により求めます。

分散 $V_x=\frac{S_{xx}}{n}$

$V_y=\frac{S_{yy}}{n}$

$V_z=\frac{S_{zz}}{n}$

相関係数 $R_{xy}=\frac{S_{xy}}{\sqrt{S_{xx}}\sqrt{S_{yy}}}$ $R_{zx}=\frac{S_{xz}}{\sqrt{S_{zz}}\sqrt{S_{xx}}}$ $R_{yz}=\frac{S_{xz}}{\sqrt{S_{yy}}\sqrt{S_{zz}}}$

■例題

次のデータより回帰分析を行います。

	1	2	3	4	5
x	24	17	22	21	20
y	2300	1500	1600	1100	900
z	9027	5983	5274	4435	3365

また、回帰係数を求めた後、次の場合のzを推定します。

$$\begin{cases} x=17 \\ y=600 \end{cases}$$

■キー操作手順

〈データ入力〉

ステップ	キー操作	表示	備考
1	DEF A	P=P−−1 P=LP−−2 _	ディスプレイ出力の選択
2	1 ENTER	X=_	データ入力
3	24 ENTER	Y=_	⋮
4	2300 ENTER	Z=_	⋮
	9027 ENTER	X=_	⋮
	⋮	⋮	⋮
		Z=_	
	3365 ENTER	X=_	
	ENTER	>	終了

〈結果出力〉

ステップ	キー操作	表示	備考
1	DEF B	P=P−−1 P=LP−−2 _	プリンタ出力の選択
2	2 ENTER	>	結果をプリンタ出力後に終了

〈推定〉

ステップ	キー操作	表示	備考
1	DEF C	P=P−−1 P=LP−−2 _	ディスプレイ出力の選択
2	1 ENTER	X=_	データ入力
3	17 ENTER	Y=_	データ入力
4	600 ENTER	Z=_ 2311.149	結果(Z)を表示
5	ENTER	X=_	
6	ENTER	>	終了

■プリント出力

(DEF B)

```
MX=       20.800
MY=     1480.000
MZ=     5616.800

VX=        5.360
VY=   233600.000
VZ=  3669666.561

RXY= 0.551005
RYZ= 0.976378
RZX= 0.481886

A=       -66.294
B=         4.043
C=      1012.543
```

(DEF A)

```
X=  24.
Y=  2300.
Z=  9027.

X=  17.
Y=  1500.
Z=  5983.

X=  22.
Y=  1600.
Z=  5274.

X=  21.
Y=  1100.
Z=  4435.

X=  20.
Y=  900.
Z=  3365.
```

(DEF C)

```
X=  17.
Y=  600.
Z=        2311.149
```

DEF A と DEF C については

プリンタ出力を選択した時のサンプルです。

■プログラムリスト

```
10:"A":REM DATA INPUT
15:CLEAR :WAIT :V=1
20:E=0
30:GOSUB 1000
40:INPUT "X=";X:GOTO 60
50:END
60:INPUT "Y=";Y,"Z=";Z
70:GOSUB 1200
80:F=F+X:G=G+Y:H=H+Z
90:I=I+X*X:J=J+Y*Y:K=K+
   Z*Z
100:L=L+X*Y:M=M+Y*Z:N=N+
    Z*X
110:E=E+1
115:GOSUB 1100
120:GOTO 40
130:END
140:"B":REM HEIKIN
150:GOSUB 1000
155:GOSUB 1300
160:X=F/E:PRINT "MX= ";X
    +W
170:Y=G/E:PRINT "MY= ";Y
    +W
180:Z=H/E:PRINT "MZ= ";Z
    +W
190:GOSUB 1100
200:O=I-E*X*X:P=J-E*Y*Y:
    Q=K-E*Z*Z
210:R=L-E*X*Y:S=M-E*Y*Z:
    T=N-E*Z*X
220:REM BUNSAN
230:GOSUB 1300
240:PRINT "VX= ";O/E+W
250:PRINT "VY= ";P/E+W
260:PRINT "VZ= ";Q/E+W
270:GOSUB 1100
280:REM SOUKAN KEISUU
290:GOSUB 1400
300:PRINT "RXY=";R/SQR (
    O*P)+W
310:PRINT "RYZ=";S/SQR (
    P*Q)+W
320:PRINT "RZX=";T/SQR (
    O*Q)+W
330:GOSUB 1100
340:REM  KAIKI KEISUU
350:GOSUB 1300
360:D=O*P-R*R
370:A=(T*P-S*R)/D:PRINT
    "A=  ";A+W
380:B=(S*O-T*R)/D:PRINT
    "B=  ";B+W
390:C=Z-A*X-B*Y:PRINT "C
    =  ";C+W
400:GOSUB 1100
410:END
```

```
420:"C":REM SUITEI
425:V=2
430:GOSUB 1000:GOSUB 130
  0
440:INPUT "X=";X:GOTO 46
  0
450:END
460:INPUT "Y=";Y
470:Z=A*X+B*Y+C
480:GOSUB 1200
485:GOSUB 1100
490:GOTO 440
500:END
1000:INPUT "P=P--1 P=LP
     --2 ";U
1010:IF U=1 PRINT =
     PRINT :GOTO 1040
1020:IF U=2 PRINT =
     LPRINT :GOTO 1040
1030:GOTO 1000
1040:RETURN
1100:REM SPACE
1110:IF U=1 GOTO 1130
1120:IF U=2 PRINT ""
1130:RETURN
1200:REM X-Y-Z PRINT
1210:IF U=1 ON V GOTO 1
     260,1240
1220:IF U=2 USING
1230:PRINT "X=  ";X:
     PRINT "Y=  ";Y
1235:IF V=1 LET W=0:
     GOTO 1250
1240:GOSUB 1300
1250:PRINT "Z=  ";Z+W
1260:RETURN
1300:REM USING 1
1310:USING "########.##
     #":W=5E-4
1320:RETURN
1400:REM USING 2
1410:USING "##.######":
     E=5E-7
1420:RETURN
1500:END
```

1119 bytes

■メモリー内容

変数	内　容
A	a
B	b
C	c
D	$Sxx \cdot Syy-(Sxy)^2$
E	n
F	$\sum xi$
G	$\sum yi$
H	$\sum zi$
I	$\sum xi^2$
J	$\sum yi^2$
K	$\sum zi^2$
L	$\sum xiyi$
M	$\sum yizi$
N	$\sum zixi$
O	Sxx
P	Syy
Q	Szz
R	Sxy
S	Syz
T	Szx
U	プリンタ出力のフラッグ
V	プログラムのフラッグ
W	四捨五入用
X	$x, \bar{x}$
Y	$y, \bar{y}$
Z	$z, \bar{z}$

PROGRAM TITLE	二元配置法

配置法とは、異なる条件のもとで得られた結果を分析し、条件と結果の間に関連があるかどうかを検定します。そのために分散分析を行い、分散比を計算します。これは、くり返しのない二元配置法のプログラムです。

■使い方

1） DEF A …データ入力及びチェック

- 要因子の水準数A、Bを入力し、続けて表示に従ってデータを入力します。
- データ入力がすべて終ると、データをチェックするかどうかを選択します。
  表示されているデータが違っている時は DEF B を押して、新しいデータを入力します。

2） DEF C …分散分析の結果出力

- 入力データもプリント出力するかどうかを選択します。
- 分散分析の結果（変動、自由度、不偏分散、不偏分散化）をプリント出力します。

■参考（計算内容など）

各水準の数：a，b

データ：$\{x_{ij}\}$　（$i=1\sim a$，$j=1\sim b$）

1. $[X]=(\sum_{i,j} x_{ij})^2/ab$
   $[A]=\sum_i(\sum_j x_{ij})^2/b$
   $[B]=\sum_j(\sum_i x_{ij})^2/a$
   $[ABS]=\sum_{i,j} x_{ij}^2$
2. $S_A=[A]-[X]$
   $S_B=[B]-[X]$
   $S_T=[ABS]-[X]$
   $S_E=[ABS]-[A]-[B]$
3. $\phi_A=a-1$
   $\phi_B=b-1$
   $\phi_E=ab-a-b-1$
   $\phi_T=ab-1$
4. $V_A=S_A/\phi_A$
   $V_B=S_B/\phi_B$
   $V_E=S_E/\phi_E$
5. $F_A=V_A/V_E$
   $F_B=V_B/V_E$

■例題

下のデータから、ホルモン投与量、飼料の種類と増体量の関連を検定するために、分散分析を行ないます。

データ：子豚のホルモン投与量を変えて、3種類の飼料を与えたときの増体量の変化〔g/日〕

		飼料の種類		
		B_1	B_2	B_3
ホルモン投与量〔mg／頭〕 A_1	10	79	68	93
ホルモン投与量〔mg／頭〕 A_2	20	86	90	100
ホルモン投与量〔mg／頭〕 A_3	30	104	110	118

■プリント出力（結果）

```
SUIJUN SUU A=3.
           B=3.
DATA
X(1,1)=79.
X(1,2)=68.
X(1,3)=93.
X(2,1)=86.
X(2,2)=90.
X(2,3)=100.
X(3,1)=104.
X(3,2)=110.
X(3,3)=118.

HENDOU (S)
 A=1432.889
 B=401.556
 E=115.111
 T=1949.556

JIYUUDO (DF)
 A=2.
 B=2.
 E=4.
 T=8.

FUHEN BUNSAN (V)
 A=716.444
 B=200.778
 E=28.778

FUHEN BUNSANHI (F)
 A=24.896
 B=6.977
```

■キー操作手順

〈データ入力とチェック〉

ステップ	キー操作	表示	備考
1	DEF A	SUIJUN A=_	Aの水準数
2	3 ENTER	SUIJUN B=_	Bの水準数
3	3 ENTER	X(1, 1)=	データ入力
		?	
4	79 ENTER	X(1, 2)=	
		?	
5	68 ENTER	X(1, 3)=	
⋮		? ⋮ 同様に入力	
11	110 ENTER	X(3, 3)=	
		?	
12	118 ENTER	INPUT END	
		CHECK(Y/N)_	入力データのチェック
13	Y ENTER	X(1, 1)=79	
14	ENTER	X(1, 2)=68	
⋮		⋮ 同様にチェック	
	ENTER	X(3, 1)=100	入力間違いを発見したので DEF B で訂正する。
19	DEF B	X(3, 1)=?	
		?	
20	104 ENTER	X(3, 1)=104	
⋮	ENTER	⋮ 続けてチェック	
23	ENTER	>	終了

〈分散分析結果の出力〉

ステップ	キー操作	表示	備考
1	DEF C	DATA PRINT(Y/N)_	入力データをプリント出力するかしないか選択。
	Y ENTER		分散分析結果をプリント出力。
		>	終了

■プログラムリスト

```
10:"A":CLEAR
20:INPUT "SUIJUN A=";A
30:INPUT "SUIJUN B=";B
40:DIM O(B-1),B$(0),X(A-1,B-1)
50:FOR I=0 TO A-1:FOR K=0 TO B-1
60:B$(0)="X("+STR$ (I+1)+","+STR$ (K+1)+")="
70:PAUSE B$(0):INPUT X(I,K):NEXT K:NEXT I:GOTO 90
80:GOTO 70
90:PAUSE "INPUT END":B$(0)=""
100:INPUT "CHECK (Y/N)=";B$(0)
110:IF B$(0)="N" END
120:IF B$(0)<>"Y" GOTO 100
125:USING
130:I=0
135:J=0
140:PRINT "X(";STR$ (I+1);",";STR$ (J+1);")=";X(I,J)
150:J=J+1:IF J=B GOTO 170
160:GOTO 140
170:I=I+1:IF I=A GOTO 190
180:GOTO 135
190:END
200:"B":PAUSE "X(";STR$ (I+1);",";STR$ (J+1);")=?"
210:INPUT X(I,J)
220:GOTO 140
230:END
240:"C":M$="":INPUT "DATA PRINT(Y/N)";M$
250:IF (M$="Y")+(M$="N")<>1 GOTO 240
260:IF M$="N" GOTO 300
270:LPRINT "":LPRINT "SUIJUN SUU A=";A:LPRINT "          B=";B
280:LPRINT "DATA":FOR I=0 TO A-1:FOR K=0 TO B-1
290:LPRINT "X(";STR$ (I+1);",";STR$ (K+1);")=";X(I,K):NEXT K:NEXT I
300:FOR J=0 TO B-1:O(J)=0:NEXT J
310:T=0:S=0:Z=0:R=0
320:FOR I=0 TO A-1:P=0:FOR J=0 TO B-1
330:E=X(I,J):Z=Z+E*E:O(J)=O(J)+E:P=P+E:NEXT J
340:S=S+P*P:R=R+P:NEXT I
350:R=R*R/(A*B):S=S/B:FOR I=0 TO B-1:T=T+O(I)*O(I):NEXT I:T=T/A
360:S=S-R:LPRINT "":LPRINT "HENDOU (S)"
365:B$(0)=" A=":C=S:GOSUB 1000
370:T=T-R:Z=Z-R:P=Z-S-T
375:B$(0)=" B=":C=T:GOSUB 1000:B$(0)=" E=":C=P:GOSUB 1000:B$(0)=" T=":C=Z:GOSUB 1000
380:F=A-1:LPRINT "":LPRINT "JIYUUDO (DF)":S=S/F:G=B-1
385:B$(0)=" A=":C=F:GOSUB 1000:B$(0)=" B=":C=G:GOSUB 1000
390:T=T/G:O=(A-1)*(B-1):P=P/O:D=A*B-1
395:B$(0)=" E=":C=O:GOSUB 1000:B$(0)=" T=":C=D:GOSUB 1000
400:LPRINT "":LPRINT "FUHEN BUNSAN (V)"
405:B$(0)=" A=":C=S:GOSUB 1000:B$(0)=" B=":C=T:GOSUB 1000:B$(0)=" E=":C=P:GOSUB 1000
410:F=S/P:LPRINT "":LPRINT "FUHEN BUNSAN HI (F)"
415:B$(0)=" A=":C=F:GOSUB 1000:B$(0)=" B=":C=T/P:GOSUB 1000
420:END
1000:REM MARUME
1010:LPRINT B$(0); INT (C*1E3+.5)/1E3
1020:RETURN
1030:END
```

1352 bytes

■メモリー内容

変　数	内　　容
A	水準数　a
B	水準数　b
C	√
D	ϕ_T
E	√
F	ϕ_a, F_a
G	ϕ_b
H	
I	√
J	√
K	√
L	
M	√
N	
O	ϕ_e
P	√, S_e, V_e
Q	
R	√, [X]
S	[A], S_a, V_a
T	[B], S_b, V_b
U	
V	
W	
X	
Y	
Z	[ABS], S_T
O(*)	$\sum_i x_{ij}^2$
X(*)	x_{ij}
B$(0)	√

CALCULATION TITLE	平均値の比較

この例は電卓モードの統計処理とBASICモードのプログラムを結びつけた例です。
標本に対応があって、各標本差d_iが、正規分布をなす母集団の標本になっていると仮定します。t値はn－1の自由度で

$$t=\frac{\bar{d}}{\frac{S_{\bar{d}}}{\sqrt{n}}}$$

で求められます。このtは2つの標本母集団が統計的に等しい平均値を持つかどうかを検定するために用います。

ただし、差の平均値$\bar{d}=\frac{\Sigma(x_i-y_i)}{n}=\frac{\Sigma x_i-\Sigma y_i}{n}$

差の標準偏差　$S\bar{d}=\left(\frac{\Sigma(x_i-y_i)^2-(\Sigma(x_i-y_i))^2/n}{n-1}\right)^{\frac{1}{2}}$

$$=\left(\frac{\Sigma x i^2-2\Sigma x_i y_i+\Sigma y i^2-n\cdot\bar{d}^2}{n-1}\right)^{\frac{1}{2}}$$

① ここでは、まず、電卓モードの統計モードでデータを入力し、U～Zのメモリーにに格納されている統計量を用いて、BASICモードでt値を求めます。

メモリー	U	V	W	X	Y	Z
統計量	Σy^2	Σy	Σxy	Σx^2	Σx	n

$\bar{d}$＝(Y－V)/Z→M (Mean：平均)
$S\bar{d}$＝SQR((X－2＊W＋U－Z＊M^2)/(Z－1))→SD
(Standard deviation：標準偏差)
t＝M/SD＊SQR Z→T（t値）
Df＝Z－1→DF (Degree of freedom：自由度)

② BASICモードで、プログラムを実行し、次のものを出力します。
SD：差の標準偏差($S\bar{d}$)
M　：差の平均（$\bar{d}$）
T　：t値（t）
DF：自由度（Df）

③ t分布表を用いて検定します。

■例題

ある学年の２つのクラス（１クラス10人）の生徒が、種々の検査により同程度の知能を持つと判断されています。この２つの集団に異なる２種類の教育方法A、Bを用いて、１年後に、検査したところ次のような結果が得られました。

A	10	4	6	2	7	13	3	11	5	9
B	8	3	4	2	3	11	4	7	4	7

これより次の仮説を検定します。

H_0：２つの教育方法は同程度に効果がある。

H_1：95％の信頼度では教育効果に差がある。

■解答例

自由度９、有意水準５％のときの仮説H_0の棄却域は $|t|>2.262$ となっています。（t分布表より求める。）

① ＢＡＳＩＣプログラムを作成する。

```
10:M=(Y-V)/Z
20:SD=SQR ((X-2*W+U-Z*M
   ^2)/(Z-1))
30:T=M/SD*SQR Z
40:DF=Z-1
50:LPRINT "SD=";SD
60:LPRINT "M =";M
70:LPRINT "T =";T
80:LPRINT "DF=";DF
90:END
```

152 bytes

② 操作

●電卓モード、統計モードを設定する。

CAL SHIFT STAT　　　　Ø

●データを入力する。

10 (x,y) 8 DATA 4 (x,y) 3 DATA …

……9 (x,y) 7 DATA　　　　10

● BASICモードにし、プログラムを実行する。

SHIFT STAT BASIC RUN ENTER

```
>
```

● 次の結果がプリントアウトされます。

```
SD=1.567021236
M =1.7
T =3.43063125
DF=9.
```

③ 結論

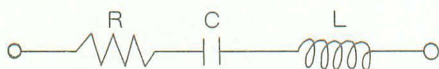
● t 値が棄却域に入るため、仮説 H_0 は棄却されます。従って教育効果に差が認められます。

PROGRAM TITLE 直列・並列回路のインピーダンス

R・L・C直列回路又は並列回路の両端に周波数 f の電流を流したときのインピーダンスを求めます。

■内容

① 直列回路



R〔Ω〕
C〔μF〕
L〔mH〕

$\omega=2\pi f$ とする。

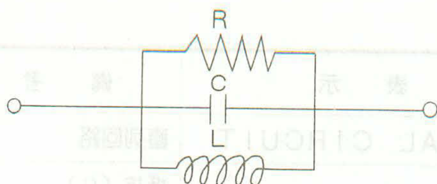
$Z=R+j(\omega L-\frac{1}{\omega C})=x+jy$

インピーダンス $|Z|=\sqrt{R^2+\left(\omega L-\frac{1}{\omega C}\right)^2}$

位相差 $\theta=\tan^{-1}\left(\frac{\omega L-\frac{1}{\omega C}}{R}\right)$

従って $(x、y)\xrightarrow{POL}(r、\theta)$ 変換により $|Z|$、θ を求める。

② 並列回路



R〔Ω〕
C〔μF〕
L〔mH〕

$\omega=2\pi f$ とする。

$$Z=\frac{1}{\frac{1}{R}+j\left(\omega C-\frac{1}{\omega L}\right)}=x+jy$$

インピーダンス $|Z|=\dfrac{1}{\sqrt{\dfrac{1}{R^2}+\left(\omega C-\dfrac{1}{\omega L}\right)^2}}$

位相差 $\theta=\tan^{-1}\left[R\left(\dfrac{1}{\omega L}-\omega C\right)\right]$

ここで $\dfrac{1}{R}+j\left(\omega C-\dfrac{1}{\omega L}\right)=x'+jy'$ とおくと次の順序で結果を得ることができます。

$$(x',\ y')\xrightarrow{\text{POL}}(r,\ \theta)\longrightarrow\left(\frac{1}{r},\ -\theta\right)\xrightarrow{\text{REC}}(x,\ y)$$

$\downarrow$ $\downarrow$

$|Z|$ θ

■使い方

1．DEF A →直列回路のインピーダンスを求めます。
抵抗 R〔Ω〕、静電容量 C〔μF〕、インダクタンス L〔mH〕そして周波数 f〔Hz〕を順に入力します。
計算結果 x、y、$|Z|$、θ〔degree〕が表示されます。

2．DEF B →並列回路のインピーダンスを求めます。
操作は DEF A と同じです。

■例題

直列回路

R＝ 5〔Ω〕
C＝10〔μF〕
L＝25〔mH〕
f＝50〔Hz〕

並列回路

R＝ 8〔Ω〕
C＝0.5〔μF〕
L＝40〔mH〕
f＝60〔Hz〕

■キー操作手順

〈直列回路〉

ステップ	キー操作	表示	備考
1	DEF A	SERIAL CIRCUIT	直列回路
		R=_	抵抗〔Ω〕
2	5 ENTER	C=_	静電容量〔μF〕
3	10 ENTER	L=_	インダクタンス〔mH〕
4	25 ENTER	F=_	周波数〔Hz〕
5	50 ENTER	X	Xの表示
6	ENTER	5	
7	ENTER	Y	Yの表示
8	ENTER	−310.4559045	
9	ENTER	IMPEDANCE Z	インピーダンス
10	ENTER	310.4961653	|Z|の表示
11	ENTER	THETA (DEG)	位相差θ (deg)の
12	ENTER	−89.07731137	の表示
13	ENTER	>	終了

〈並列回路〉

ステップ	キー操作	表示	備考
1	DEF B	PARALLEL CIRCUT	並列回路
		R=_	抵抗〔Ω〕
2	8 ENTER	C=_	静電容量〔μF〕
3	0.5 ENTER	L=_	インダクタンス〔mH〕
4	40 ENTER	F=_	周波数〔Hz〕
5	60 ENTER	X	Xの表示
6	ENTER	6.250732478	
7	ENTER	Y	Yの表示
8	ENTER	3.306690691	
9	ENTER	IMPEDANCE Z	インピーダンス
10	ENTER	7.071482153	|Z|の表示
11	ENTER	THETA (DEG)	位相差θ (deg)
12	ENTER	27.87922142	の表示
13	ENTER	>	終了

■プログラムリスト

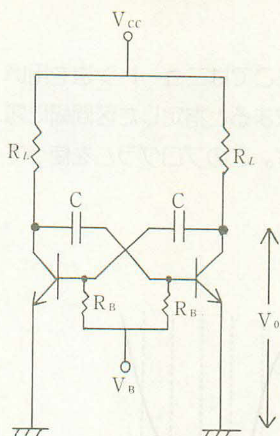
```
 10:"A":A=0:PAUSE "SERIA
    L CIRCUIT":GOTO 30
 20:"B":A=1:PAUSE "PARAL
    LEL CIRCUIT"
 30:DEGREE :WAIT
 40:INPUT "R=";R
 50:INPUT "C=";C:C=C*1E-
    6
 60:INPUT "L=";L:L=L*1E-
    3
 70:INPUT "F=";F
 80:W=2*π*F
 90:IF A=1 GOTO 250
100:REM SERIAL
110:U=R:V=W*L-RCP (W*C)
120:Y=POL (U,V)
150:REM PRINT
160:PRINT "X":PRINT U
170:PRINT "Y":PRINT V
180:PRINT "IMPEDANCE Z":
    PRINT Y
190:PRINT "THETA(DEG)":
    PRINT Z
200:END
250:REM PARALLEL
260:Y=RCP R:Z=W*C-RCP (W
    *L)
270:Y=POL (Y,Z):U=RCP Y
280:V=-Z
290:Y=REC (U,V)
300:REM PRINT
310:PRINT "X":PRINT Y
320:PRINT "Y":PRINT Z
330:PRINT "IMPEDANCE Z":
    PRINT U
340:PRINT "THETA(DEG)":
    PRINT V
350:END
```

426 bytes

■メモリー内容

変数	内容	
A	フラッグ	
B		
C	静電容量	
D		
E		
F	周波数	
G		
H		
I		
J		
K		
L	インダクタンス	
M		
N		
O		
P		
Q		
R	抵抗値	
S		
T		
U	x	$\|Z\|$
V	y	θ
W	$2\pi f\ (w)$	
X		
Y	$\|z\|$	x', x
Z	θ	y', y

CALCULATION TITLE　パルス回路



左図のような非安定マルチバイブレータ回路には次の様な関係式が成立します。

$$T = C R_B\, ln\left(\frac{V_{CC} + V_B}{V_{CC}}\right)$$

$$V_0 = V_{CC}\left\{1 - \exp\left(-\frac{1}{C R_L} t\right)\right\}$$

$$t_r = -C R_L \{ln(1-0.9) - ln(1-0.1)\}$$

T：周期　　t_r：立上り時間

■例題

上式を利用して次の様な場合の周期・立上り時間を求めなさい。

V_{CC}＝12〔V〕　V_B＝9〔V〕　R_L＝1〔kΩ〕

C＝500〔pF〕 R_B＝50〔kΩ〕

■操作方法

① 電卓モードの設定・浮動小数点表示の指示

② 周期Tを求めます。

500 [EXP] 12 [+/−] [×] 50 [EXP] 3 [×]　　　0.000025

[(] [(] 12 [+] 9 [)] [÷] 12 [)] [ln] [=]　　　0.00001399

[F↔E]（表示方式の切替え）　　　1.39903947E−05

答　約14.0〔μs〕

① 立上り時間T_rを求めます。

500 [+/−] [EXP] 12 [+/−] [×] [EXP] 3 [×] [(] [(]

1 [−] 0.9 [)] [ln] [−] [(] 1 [−] 0.1

[)] [ln] [)] [=]　　　0.000001098

[F↔E]（表示方式の切替え）　　　1.098612289E−06

答　約1.1〔μs〕

PROGRAM TITLE	方程式の根(ニュートン法)

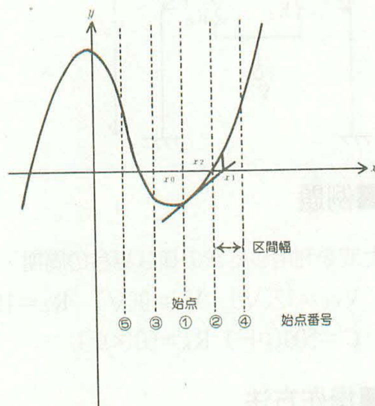
■簡単明解、根の算出！

一般に方程式の根を求めるのは、やっかいなものですが、ここではニュートン法を用いて根を近似して行く方法をとり上げています。1ツの根が求まると指定した区間幅に応じてニュートン法を用いる始点が自動的に変わって行きます。このプログラムを使って面倒な方程式の根もかんたんに求めることができます。

■使い方

1．DEF A でプログラム開始です。
2．始点、微小区間、区間幅をそれぞれ入力します。
3．根の値を出力します。さらにENTERを繰り返し押すことにより、始点を先に入力した区間幅ごとに変えていって根を求めます。

注）このプログラムはニュートン法の基礎的なアルゴリズムであり、複数個の解があっても一部解が表示されないことがあります。



■参考(計算内容など)

関数$y=f(x)$の$x=Xn$における接線がx軸と交わる時のxの値をX_{n+1}とする

$$X_{n+1}=Xn-\frac{f(Xn)}{f'(Xn)}$$

となります。

XnとX_{n+1}の差の絶対値が10^{-8}以下になればXnを根とみなし表示します。ここでは1次微分の$f'(X)$を次の様に定義します。

$$f'(X)=\frac{f(X+h)-f(X)}{h}\quad(hは微小区間)$$

10^{-8}を変更する時は、340ラインの1E−8を変更してください。

■例　題

$x^3-2x^2-x+2=0$（根は−1，1，2）

始点=0

微小区間$=10^{-4}$

区間幅=0.5

として計算します。

関数は、サブルーチンとして、500ライン以後に書き込んでください。

書き込み方（例題の場合）

1．PROモードにする。

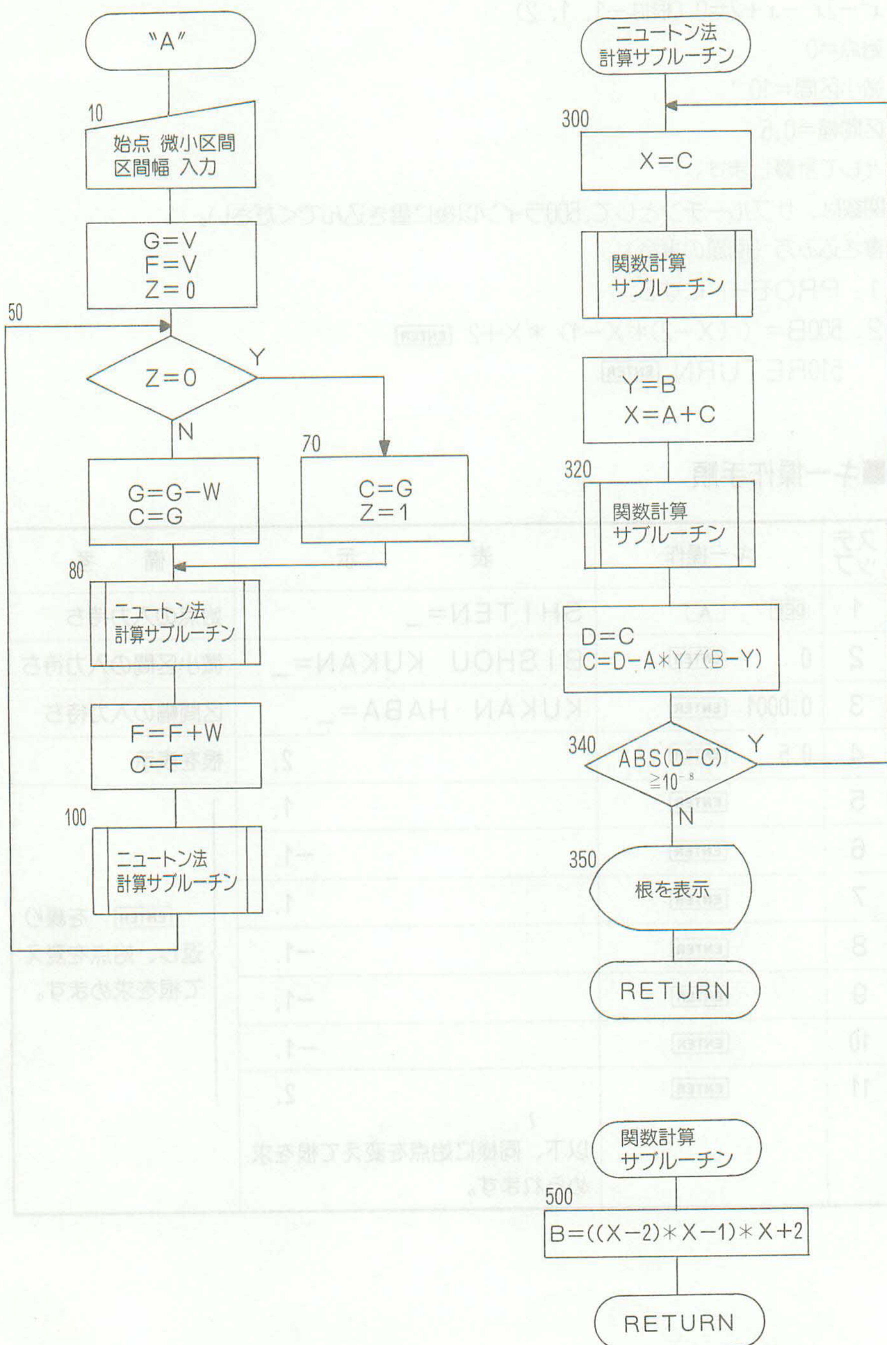
2．500B=（（X−2)＊X−1）＊X+2 ENTER

510RETURN ENTER

■キー操作手順

ステップ	キー操作	表示	備考
1	DEF　A	SHITEN=_	始点の入力待ち
2	0　ENTER	BISHOU KUKAN=_	微小区間の入力待ち
3	0.0001　ENTER	KUKAN HABA=_	区間幅の入力待ち
4	0.5　ENTER	2.	根を表示
5	ENTER	1.	ENTER を繰り返し、始点を変えて根を求めます。
6	ENTER	−1.	
7	ENTER	1.	
8	ENTER	−1.	
9	ENTER	−1.	
10	ENTER	−1.	
11	ENTER	2. 〜 以下、同様に始点を変えて根を求められます。	

■フローチャート


"A"
10
始点 微小区間
区間幅 入力
G=V
F=V
Z=0
50
Z=0
Y
N
G=G−W
C=G
70
C=G
Z=1
80
ニュートン法
計算サブルーチン
F=F+W
C=F
100
ニュートン法
計算サブルーチン
ニュートン法
計算サブルーチン
300
X=C
関数計算
サブルーチン
Y=B
X=A+C
320
関数計算
サブルーチン
D=C
C=D−A*Y/(B−Y)
340
ABS(D−C)
≧10⁻⁸
Y
N
350
根を表示
RETURN
関数計算
サブルーチン
500
B=((X−2)*X−1)*X+2
RETURN

■プログラムリスト

```
 10:"A":INPUT "SHITEN=";
    V
 20:INPUT "BISHOU KUKAN=
    ";A
 30:INPUT "KUKAN HABA=";
    W
 40:G=V:F=V:Z=0
 50:IF Z=0 GOTO 70
 60:G=G-W:C=G:GOTO 80
 70:C=G:Z=1
 80:GOSUB 300
 90:F=F+W:C=F
100:GOSUB 300
110:GOTO 50
120:END
300:X=C:GOSUB 500
310:Y=B:X=A+C
320:GOSUB 500
330:D=C:C=D-A*Y/(B-Y)
340:IF ABS (D-C)>=1E-8
    GOTO 300
350:BEEP 3:PRINT C
360:RETURN
500:B=((X-2)*X-1)*X+2
510:RETURN
```

272 bytes

■メモリ内容

変数	内容
A	微小区間
B	f(X)
C	X_0
D	f(x+h)
E	
F	√
G	√
H	
I	
J	
K	
L	
M	
N	
O	
P	
Q	
R	
S	
T	
U	
V	始点
W	区間幅
X	x
Y	f(x)
Z	初期フラッグ

PROGRAM TITLE

数値積分(シンプソン)

■シンプル算出、シンプソン積分

このプログラムでは、積分区間内で等間隔幅に与えられた関数値に対し、数値積分を行ないます。また、関数式をプログラムに組み込んでおけば積分区間内の各値を自動計算して積分します。

■使い方

1. DEF A　操作→初期値入力

   データを入力しての計算か、関数式を組み込んでの計算かの選択を行ないます。積分区間の始点、終点、分割数も入力します。関数式を組み込む場合は、プログラムの900ラインに関数式を書き込みます。(書き込み方参照)

2. DEF B　操作→訂正値入力

   プリント出力のデータ番号を入力した後、訂正データを入力します。さらにすべてのデータをプリントするか、プリントしないか選択し処理終了です。

3. DEF C　操作→積分値出力

   積分区間の各データ（関数値）をもとに、積分値がプリントされます。

   注）分割数は2～254までの偶数を入力してください。

■例　　題

1．次のデータを入力して計算を行ないます。

区間(0,5)．分割数40

$f(x_1)$	4	$f(x_{12})$	−7	$f(x_{23})$	−2	$f(x_{34})$	13
$f(x_2)$	5.5	$f(x_{13})$	−8	$f(x_{24})$	0	$f(x_{35})$	12 .5
$f(x_3)$	6	$f(x_{14})$	−9	$f(x_{25})$	2	$f(x_{36})$	12
$f(x_4)$	5.7	$f(x_{15})$	−9.5	$f(x_{26})$	4	$f(x_{37})$	10 .5
$f(x_5)$	5	$f(x_{16})$	−10	$f(x_{27})$	6	$f(x_{38})$	9
$f(x_6)$	2	$f(x_{17})$	−9.5	$f(x_{28})$	7	$f(x_{39})$	7.8
$f(x_7)$	0	$f(x_{18})$	−9	$f(x_{29})$	8	$f(x_{40})$	6
$f(x_8)$	−1.8	$f(x_{19})$	−8.5	$f(x_{30})$	9.7	$f(x_{41})$	4
$f(x_9)$	−3	$f(x_{20})$	−7	$f(x_{31})$	11		
$f(x_{10})$	−5	$f(x_{21})$	−5.5	$f(x_{32})$	12		
$f(x_{11})$	−6	$f(x_{22})$	−4	$f(x_{33})$	12 .5		

2．関数式を組み込んで計算します。

$y=\{(x-2)x-1\}\ x+2$

区間〔0，1〕分割数20

関数はサブルーチンとして900ライン以後に書き込んでください。

■書き込み方（例題の場合）

PROモードにして

900Y=((X−2)＊X−1)＊X+2 ENTER

910RETURN ENTER　で書き込み終了

■プリント出力（結果）

```
A=0.
B=5.
N=40.
  1 F(0)=4.
  2 F(0.125)=5.5
  3 F(0.25)=6.
  4 F(0.375)=5.7
  5 F(0.5)=5.
  6 F(0.625)=2.
  7 F(0.75)=0.
  8 F(0.875)=-1.8
  9 F(1)=-3.
 10 F(1.125)=-5.
 11 F(1.25)=-6.
 12 F(1.375)=-7.
 13 F(1.5)=-8.
 14 F(1.625)=-9.
 15 F(1.75)=-9.5
 16 F(1.875)=-10.
 17 F(2)=-9.5
 18 F(2.125)=-9.
 19 F(2.25)=-8.5
 20 F(2.375)=-7.
 21 F(2.5)=-5.5
 22 F(2.625)=-4.
 23 F(2.75)=-2.
 24 F(2.875)=0.
 25 F(3)=2.
 26 F(3.125)=4.
 27 F(3.25)=6.
 28 F(3.375)=7.
 29 F(3.5)=8.
 30 F(3.625)=9.7
 31 F(3.75)=11.
 32 F(3.875)=12.
 33 F(4)=12.5
 34 F(4.125)=13.
 35 F(4.25)=12.5
 36 F(4.375)=12.
 37 F(4.5)=10.5
 38 F(4.625)=9.
 39 F(4.75)=7.8
 40 F(4.875)=6.
 41 F(5)=4.

F=8.29166667
```

```
A=0.
B=1.
N=20.

  1 F(0)=2.
  2 F(0.05)=1.945125
  3 F(0.1)=1.881
  4 F(0.15)=1.808375
  5 F(0.2)=1.728
  6 F(0.25)=1.640625
  7 F(0.3)=1.547
  8 F(0.35)=1.447875
  9 F(0.4)=1.344
 10 F(0.45)=1.236125
 11 F(0.5)=1.125
 12 F(0.55)=1.011375
 13 F(0.6)=0.896
 14 F(0.65)=0.779625
 15 F(0.7)=0.663
 16 F(0.75)=0.546875
 17 F(0.8)=0.432
 18 F(0.85)=0.319125
 19 F(0.9)=0.209
 20 F(0.95)=0.102375
 21 F(1)=0.

F=1.083333333
```

■キー操作手順

〈データを入力しての計算〉

ステップ	キー操作	表示	備考
1	DEF A	INP. =1/CAL. =2?_	データを入力するか組み込んだ計算式をもとに計算するかの選択
2	1 ENTER	A=_	積分区間始点の入力待ち
3	0 ENTER	B=_	積分区間終点の入力待ち
4	5 ENTER	N=_	積分区間内分割数の入力待ち
5	40 ENTER	F(0)= ?	データの入力待ち
6	4 ENTER	F(0.125)= ?	データの入力待ち
7	5.5 ENTER	F(0.25)= ?	データの入力待ち
		～	
		以下同様にデータの入力を行います。	
45	6 ENTER	F(5)= ?	データの入力待ち
46	4 ENTER	＞	データ入力の終了
47	DEF B	TEISEI NO. ?=_	訂正するデータ番号の入力待ち
48	3 ENTER	TEISEI CHI ?=_	訂正値の入力待ち
49	6 ENTER	TEISEI NO. ?=_	訂正するデータ番号の入力待ち

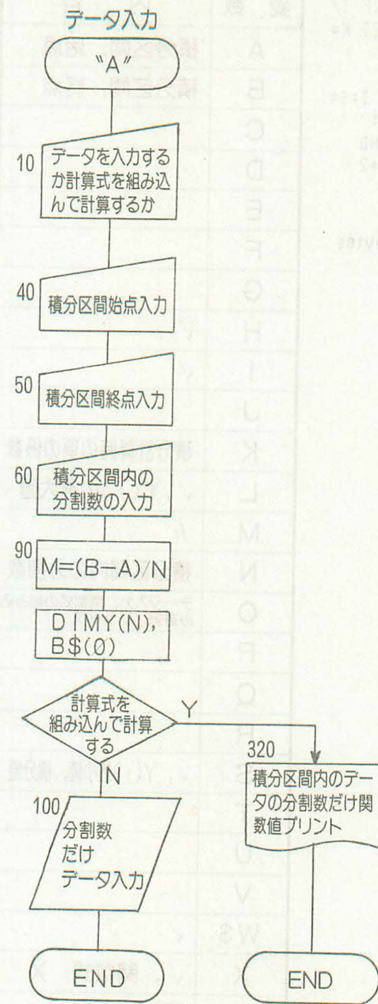
ステップ	キー操作	表示	備考
50	ENTER	ALL PRINT(Y/N)_	データ訂正の終了すべてのデータをプリントするかの選択
51	Y ENTER	〉	すべてのデータをプリントして処理終了
52	DEF C	〉	積分値をプリントして処理終了

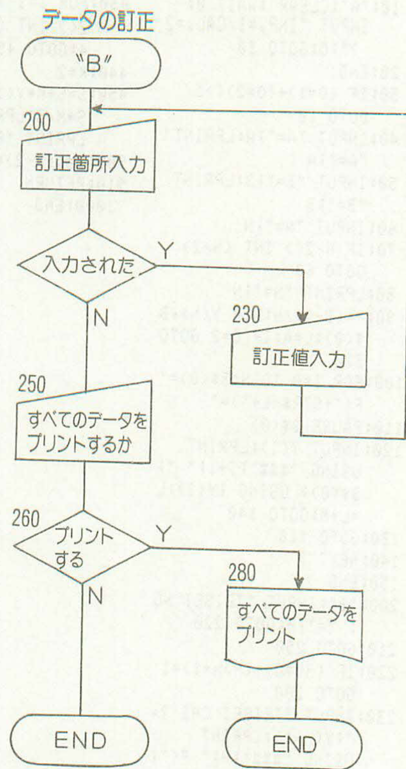
■キー操作手順

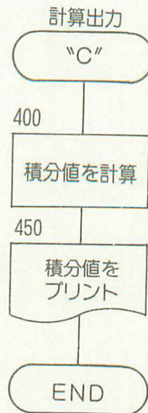
〈関数式を組み込んでの計算〉

ステップ	キー操作	表示	備考
1	DEF A	INP. =1/CAL. =2?_	データを入力するか組み込んだ計算式をもとに計算するかの選択
2	2 ENTER	A=_	積分区間始点の入力待ち
3	0 ENTER	B=_	積分区間終点の入力待ち
4	1 ENTER	N=_	積分区間内分割数の入力待ち
5	20 ENTER	〉	関数式をもとに関数値をプリントして処理終了
6	DEF C	〉	積分値をプリントして処理終了

■フローチャート


データ入力
"A"
10
データを入力するか計算式を組み込んで計算するか
40
積分区間始点入力
50
積分区間終点入力
60
積分区間内の分割数の入力
90
M=(B−A)/N
DIMY(N),B$(0)
計算式を組み込んで計算する
Y
N
100
分割数だけデータ入力
320
積分区間内のデータの分割数だけ関数値プリント
END
END


データの訂正
"B"
200
訂正箇所入力
入力された
Y
N
230
訂正値入力
250
すべてのデータをプリントするか
260
プリントする
Y
N
280
すべてのデータをプリント
END
END


計算出力
"C"
400
積分値を計算
450
積分値をプリント
END

■プログラムリスト

```
10:"A":CLEAR :WAIT 0:
   INPUT "INP.=1/CAL.=2
   ?";O:GOTO 30
20:END
30:IF (O=1)+(O=2)<>1
   GOTO 10
40:INPUT "A=";A:LPRINT
   "A=";A
50:INPUT "B=";B:LPRINT
   "B=";B
60:INPUT "N=";N
70:IF N/2<> INT (N/2)
   GOTO 60
80:LPRINT "N=";N
90:M=(B-A)/N:DIM Y(N),B
   $(0):L=A:IF O=2 GOTO
   320
100:FOR I=0 TO N:B$(0)="
   F("+STR$ L+")="
110:PAUSE B$(0)
120:INPUT Y(I):LPRINT
   USING "###";I+1;" ";
   B$(0); USING ;Y(I):L
   =L+M:GOTO 140
130:GOTO 110
140:NEXT I
150:END
200:"B":INPUT "TEISEI NO
   . ?=";H:GOTO 220
210:GOTO 250
220:IF (H<=0)+(H>N+1)=1
   GOTO 200
230:INPUT "TEISEI CHI ?=
   ";Y(H-1):LPRINT
   USING "###";H;" F(";
   STR$ (A+M*(H-1));")=
   "; USING ;Y(H-1):
   GOTO 200
240:GOTO 230
250:INPUT "ALL PRINT (Y/
   N)";W$
260:IF (W$="Y")+(W$="N")
   <>1 GOTO 250
270:IF W$="N" END
280:LPRINT "":LPRINT "A=
   ";A:LPRINT "B=";B:
   LPRINT "N=";N
290:M=(B-A)/N:L=A:FOR I=
   0 TO N
300:LPRINT USING "###";(
   I+1);" F(";STR$ L;")
   ="; USING ;Y(I)
310:L=L+M:NEXT I:END
320:LPRINT "":FOR I=0 TO
   N:X=L:GOSUB 900
330:LPRINT USING "###";(
   I+1);" F(";STR$ L;")
   ="; USING ;Y
340:Y(I)=Y:L=L+M:NEXT I:
   END
400:"C":S=Y(0):L=S:FOR I
   =0 TO N:IF S>Y(I)
   LET S=Y(I)
410:NEXT I
420:S=Y(0)+Y(N)
430:FOR I=1 TO N-1:IF I/
   2<> INT (I/2) LET K=
   4:GOTO 450
440:K=2
450:S=S+K*Y(I):NEXT I:S=
   S*M/3:LPRINT "":
   LPRINT "F=";S:END
900:Y=((X-2)*X-1)*X+2
910:RETURN
1000:END
```

859 bytes

■メモリ内容

変　数	内　　容
A	積分区間，始点
B	積分区間，終点
C	
D	
E	
F	
G	
H	√
I	√
J	
K	積分計算時の項の係数
L	√，Y(i)の最大値
M	h
N	積分区間内の分割数
O	データ入力，関数式の組み込み選択，フラッグ
P	
Q	
R	
S	√，Y(i)最小値，積分値
T	
U	
V	
W$	√
X	√，関数式　X
Y	関数式　Y
Z	
Y(N)	入力データ(関数値)
B$(0)	√

PROGRAM TITLE	軟着陸

燃料が少ない！緊急着陸せよ！！

残り少ない燃料でいかにロケットを着陸させるか。ロケットは自由落下する。それに合わせてロケットエンジンを点火せよ。点火するのが早かったり、燃料消費が多過ぎると惑星の塵となってしまい、遅過ぎたりすると惑星面に墜突してしまう。さあ、あなたの腕と度胸でこのロケットを無事に着陸させてみよう。点火時と燃料消費のコントロールが成功させるコツだよ。

■遊び方

1. DEF A でプログラムをスタートします。
2. 0 ～ 9 までのキーで消費燃料を入力します。
   着陸させる時にはうまく燃料を調整しながら着陸させてください。成功した時は残り燃料、失敗した時は "GOOD　BY!!" が表示され、再ゲームをするかどうかの入力待ちとなります。また、燃料がなくなった時は、落下中「FALLING」の表示が出ます。

■参考

重力は5m/s²単位時間(S)に5単位の燃料を消費すると重力を相殺します。

計算式

$H=H_0+V_{0t}+\frac{1}{2}at^2$

$V=V_0+at$

$V^2=V_0^2+2aH$

$H_0=500$、$V_0=-50$、$F_0=200$

H：高度
V：速度
a：重力加速度
t：時間
V_0：初速度
H_0初高度
F_0：初燃料

注）プログラムの30ラインの初高度、初燃料とwait 時間の数値をかえることによって面白さをかえることができます。

■キー操作手順

ステップ	キー操作	表示	備考
1	DEF A	＊＊＊START＊＊＊	ゲーム開始
2	9	500 −50 200 0	高度・速度・残り燃料・消費燃料の表示
3	9	452 −46 191 9	
		～ 操り返します．	
		SEIKO!!	成功の場合
		NENRYO F=10.	残り燃料の表示
		REPLAY(Y/N)?	再ゲームするかどうか選択
31	Y		ステップ1へもどる
	N	>	プログラム終了

■プログラムリスト

```
 10:"A":WAIT 50:CLEAR :
    USING :S=-50:A=0:D$=
    ""
 20:BEEP 3:PRINT "  ***
    START ***"
 30:DATA "WAIT=",50,"NEN
    RYOU=",200,"KODO=",5
    00
 40:RESTORE
 50:READ B$,W,B$,F,B$,H
 60:WAIT W
 70:IF F=0 IF S<0 PRINT
    USING "####";H;S;" F
    ALLING":GOTO 90
 80:PRINT USING "####";H
    ;S;F;C
 90:BEEP 1:D$="":D$=
    INKEY$
100:IF D$="" LET C=A:
    GOTO 130
110:C=VAL D$
120:A=C
130:IF C>F LET C=F
140:F=F-C:X=C-5:H=H+S+X/
    2:S=S+X
150:IF H>5 GOTO 70
160:IF H>0 AND ABS S<5
    BEEP 5:PRINT "SEIKO!
    !":GOTO 180
170:BEEP 3:PRINT "GOOD B
    Y !!":GOTO 190
180:WAIT 150:PRINT USING
    "####";"NENRYO  F=";
    F
190:WAIT 50:PRINT "REPLA
    Y (Y/N) ?":Z$=INKEY$
200:IF Z$="Y" OR Z$="N"
    GOTO 220
210:GOTO 190
220:IF Z$="Y" GOTO 10
230:END
```

461 bytes

■メモリー内容

変数	内容
A	√
B$	√
C	消費燃料
D$	消費燃料
E	
F	初燃料、残り燃料
G	
H	初高度、高度
I	
J	
K	
L	
M	
N	
O	
P	
Q	
R	
S	速　度
T	
U	
V	
W	WAIT時間
X	√
Y	
Z$	√

PROGRAM TITLE	タイプ練習

すばやく、正しく、タイプイン！

ポケットコンピュータのキー配列は、タイプライターと同じ。英文タイプを経験していると比較的なじみやすいものですが、そうでない人はどのキーがどこにあるのやら、すぐに覚えられないものです。そこでこのプログラムがお手伝い。コンピュータが出題する文字を制限時間内にすばやくキー入力して、得点を競うゲームです。高得点を出してキー入力のテクニックを磨いてください。

■遊び方

1. DEF Z でプログラム開始です。グレード（１．２．３）を選択し入力すると４～６文字のアルファベットが表示されますから、制限時間内にキーボードを操作し、入力してください。
2. 10問終了後、スコアが表示されます。各問題で、全部正解の場合10点、半分以上正解は５点で合計得点に加算されます。制限時間は、グレード３が最短で、グレード１が最長です。
3. ゲームは同じグレード内で得点を競ってください。問題は10問、最高得点は100点です。
4. DEF A は、グレードは同じで実行したいときに使ってください。

■キー操作手順

ステップ	キー操作	表　　示	備　考
1	DEF Z	GRADE(1, 2, 3)?_	グレードの入力待ち
2	1 ENTER	AZBCD	
3	A	AZBCD A	
4	Z	AZBCD AZ	
⁀		⁀ 以下、同様に行います。	
		YOUR-SCORE=80	10問終了後 スコア表示
		YOUR SCORE BEST	自己のスコアがハイスコアより高い場合の表示
		>	終了
1	DEF A	HIGH-SCORE=80.	同じグレードで実行したい時
		BWVS	
2	B	BWVS B	
⁀		⁀ 以下、同様に行います。	
		YOUR-SCORE=60.	
		>	終了

■プログラムリスト

```
 10:"Z":CLEAR :DIM B$(5)
   ,C$(5):RANDOM
 15:INPUT "GRADE(1,2,3)?
   ";L:WAIT 0
 17:IF (L=1)+(L=2)+(L=3)
   <>1 THEN 15
 18:GOTO 30
 20:"A":WAIT 0:P=0:PAUSE
   "HIGH-SCORE=";X
 30:FOR S=1 TO 10
 40:B=RND 4+2:Y$="":R=
   INT (B/2)
 50:FOR C=0 TO B-1:C$(C)
   =""
 60:D=RND 26:B$(C)=CHR$
   (D+&40):Y$=Y$+CHR$ (
   D+&40):NEXT C:A$=""
 70:BEEP 3:E=0:WAIT 30:
   USING "&&&&&&&&"
 80:FOR W=1 TO B*10/L:
   PRINT Y$;A$:IF E=B
   LET W=B*20/L:GOTO 10
   0
 85:C$(E)=INKEY$ :IF C$(
   E)="" THEN 100
 87:A$=A$+C$(E)
 90:E=E+1
100:NEXT W:Q=0
110:FOR W=0 TO B-1:IF B$
   (W)=C$(W) LET Q=Q+1
120:NEXT W:IF Q<=R THEN
   150
130:IF Q=B LET P=P+10:
   GOTO 150
140:P=P+5
150:NEXT S:USING :BEEP 3
   :PAUSE "YOUR-SCORE="
   ;P
160:IF P>X LET X=P:WAIT
   100:PRINT "YOUR SCOR
   E BEST"
170:END
```

491 bytes

■メモリ内容

変　数	内　　容
A$	√
B	√
C	ループカウンター
D	√
E	√
F	
G	
H	
I	
J	
K	
L	グレード
M	
N	
O	
P	スコア
Q	√
R	√
S	ループカウンター
T	
U	
V	
W	ループカウンター
X	ハイスコア
Y$	√
Z	
B$(5)	√
C$(5)	√

PROGRAM TITLE	集計票

伝票集計などにとっても便利な簡易集計プログラムです。コード又は項目名をランダムに入力し、各コード（項目）ごとに合計、構成比を求め、最後に総合計をプリントします。

■操作方法

1. DEF Z （項目名の登録）

   コード番号順に、項目名を登録します。

2. DEF A （集計データ入力）

   コード番号を入力すると項目名が表示されます。続けて集計データを入力します。
   コード番号がわからない時は“＊”の後に項目名を入力すると、コード番号を検索します。

3. DEF S （集計結果のプリント出力）

   各コード（項目）ごとの集計を行い、集計結果と構成比をプリントします。
   最初に構成比の要・不要を入力します。

■例題

コード表

コード	項目名
1	PC−1245
2	PC−1251
3	PA−7050
4	EL−331
5	EL−332
6	WN−106
7	CT−660
8	EL−550
9	CS−2302
10	EA−11E
11	EA−150
12	EA−850C

コード	単価	数量	金額
4	5,800	3	
3			39,800
8	14,800	15	
9	19,800	20	
11	3,300	40	
12	800	18	
8	14,800	20	
7			178,000
8			74,000
3			597,000
12	800	99	
2	29,800	4	

■参考

必要に応じて10ラインの配列C$(*)、D(*)の大きさはかえてください。

プログラムリストでは項目数は50に設定してあります。

■キー操作手順

〈項目名の登録〉

ステップ	キー操作	表示	備考
1	DEF Z	1	
		KOUMOKU=_	コードNO.1の項目名入力待ち
2	PC−1245 ENTER	2	
		KOUMOKU=_	コードNO.2の項目名入力待ち
3	PC−1251 ENTER	3	
	⋮	⋮ (以下同様に入力します) ⋮	
13	EA−850C ENTER	13	
		KOUMOKU=_	
14	ENTER	>	終了

〈集計データ入力〉

ステップ	表　示	表　　　示	備　　考
1	DEF　A	SYUUKEI	タイトル表示
		CODE=_	
2	4 ENTER	EL-331	
		DATA=_	
3	5800 ＊ 3 ENTER	CODE=_	
4	3 ENTER	PA-7050	
		DATA=_	
5	39800 ENTER	CODE=_	
6	＊ EL-550 ENTER	8	コード番号不明時の入力方法
		DATA=_	
		⋮ (以下同様に入力をくり返します。) ⋮	
25	29800 ＊ 4 ENTER	CODE=_	
26	ENTER	>	終了

〈集計結果のプリント出力〉

ステップ	キー操作	表　　　示	備　　考
1	DEF　S	SYUUKEI　LIST	タイトル表示
		KOUSEIHI(Y/N)_	
	Y ENTER	>	●集計結果プリント出力(構成比も含む)
	N ENTER	>	●集計結果プリント出力（構成比は含まない)

■プリント出力

●構成比を含む場合

```
 1   PC-1245
           0    0.00%
 2   PC-1251
      119200    5.51%
 3   PA-7050
      636800   29.41%
 4   EL-331
       17400    0.80%
 5   EL-332
           0    0.00%
 6   WN-106
           0    0.00%
 7   CT-660
      178000    8.22%
 8   EL-550
      592000   27.34%
 9   CS-2302
      396000   18.29%
10   EA-11E
           0    0.00%
11   EA-150
      132000    6.10%
12   EA-850C
       93600    4.32%

TOTAL         2165000
```

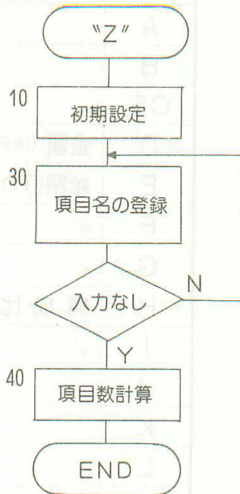
●構成比を含まない場合

```
 1   PC-1245
                 0
 2   PC-1251
            119200
 3   PA-7050
            636800
 4   EL-331
             17400
 5   EL-332
                 0
 6   WN-106
                 0
 7   CT-660
            178000
 8   EL-550
            592000
 9   CS-2302
            396000
10   EA-11E
                 0
11   EA-150
            132000
12   EA-850C
             93600

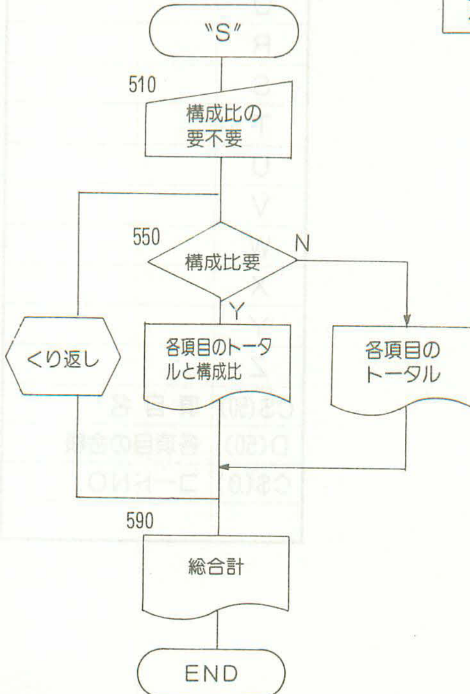
TOTAL      2165000
```

■フローチャート

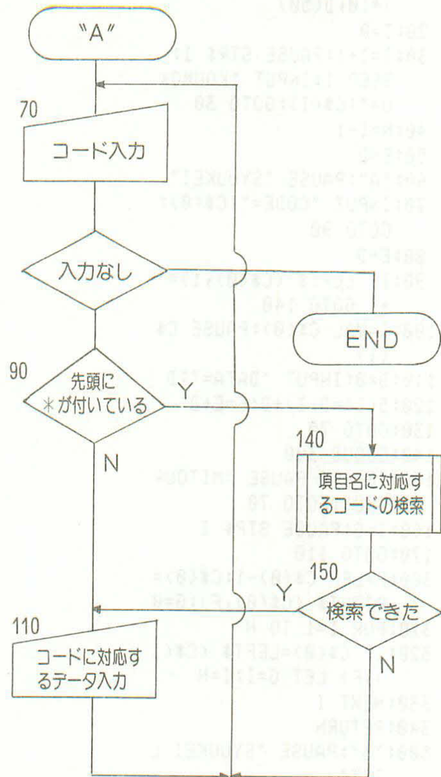
項目名の登録



集計リスト



集　計



■プログラムリスト

```
10:"Z":CLEAR :DIM C$(50
   )*10,D(50)
20:I=0
30:I=I+1:PAUSE STR$ I:
   BEEP 1:INPUT "KOUMOK
   U=";C$(I):GOTO 30
40:N=I-1
50:END
60:"A":PAUSE "SYUUKEI"
70:INPUT "CODE=";C$(0):
   GOTO 90
80:END
90:IF LEFT$ (C$(0),1)="
   *" GOTO 140
100:I=VAL C$(0):PAUSE C$
    (I)
110:D=0:INPUT "DATA=";D
120:D(I)=D(I)+D:E=E+D
130:GOTO 70
140:GOSUB 300
150:IF G=0 PAUSE "MITOUR
    OKU":GOTO 70
160:I=G:PAUSE STR$ I
170:GOTO 110
300:F=LEN C$(0)-1:C$(0)=
    RIGHT$ (C$(0),F):G=0
310:FOR I=1 TO N
320:IF C$(0)=LEFT$ (C$(I
    ),F) LET G=I:I=N
330:NEXT I
340:RETURN
500:"S":PAUSE "SYUUKEI L
    IST"
510:INPUT "KOUSEIHI(Y/N)
    ";C$
520:IF (C$="Y")+(C$="N")
    <>1 GOTO 510
530:FOR I=1 TO N
540:LPRINT USING "###";I
    ;"   ";C$(I)
550:IF C$="N" LPRINT "
           "; USING "#
    ##########";D(I):
    GOTO 580
560:H=D(I)/E*100:H= INT
    (H*100+.5)/100
570:LPRINT "    "; USING
    "###########";D(I);"
     "; USING "###.##";H
    ;"%"
580:NEXT I
590:LPRINT "":LPRINT "TO
    TAL       "; USING "#
    ##########";E
600:END
```

648 bytes

■メモリー内容

変　数	内　容
A	
B	
C$	√
D	金額 (INPUT DATA)
E	総額(TOTAL)
F	√
G	√
H	構 成 比
I	√
J	
K	
L	
M	
N	項 目 数
O	
P	
Q	
R	
S	
T	
U	
V	
W	
X	
Y	
Z	
C$(50)	項 目 名
D(50)	各項目の金額
C$(0)	コードNO.

CALCULATION TITLE	割賦計算

高額商品は、一般の人々にとっては、買いたくても手が出にくいものです。
そこで登場するのが割賦計算です。

$$P=(PT-R)\times\frac{i}{1-\frac{1}{(1+i)^n}}$$

P：割賦金
PT：定価
R：頭金
n：返済回数
i：月利

■例題

550,000円の工作機械を頭金100,000円、月利1.5%、20回の月賦払いで購入すると、毎月の返済額は、どれほどに見積れば良いでしょうか。

■解答

PT＝550,000円
R＝100,000円
n＝20回
i＝1.5%＝0.015

割賦金を求めます。

[( ] 550000 [−] 100000 [ )] [×] 0.015
[÷] [( ] 1 [−] [( ] 1 [+] 0.015 [ )] [y^x] 20 [+/−] [ )] [=]

26210.58114

従って毎月の返済額は約26,210円となります。

エラーコードとエラー内容

エラーコード	エ ラ ー 内 容
1	文法エラー ●文法的に実行できない場合のエラー 例 3＊／2 ENTER ●プログラムのみで実行可能な命令をマニュアル操作で、マニュアル操作でのみ実行可能な命令をプログラムで実行しようとした 例 PRINT A ENTER （マニュアル操作）
2	演算エラー ●計算結果が1×10^{100}以上になった ●除数が0の除算を実行 ●不合理な演算の実行（LN−30、ASN1.5など） ●その他
3	DIMエラー、範囲制限エラー ●すでに宣言されている配列変数名を再度宣言している。 DIM命令で宣言していない配列変数を使っている。 DIM命令で宣言した大きさを越えて配列変数を使っている など ●指定の数値が規定されている範囲から外れている。 例 DIM B（256） ENTER （配列変数の配列要素が255を越えている。） 10 WAIT 66000 （停止時間の指定が65535を越えている）
4	ラインエラー ●GOTO、GOSUB、RUN、LIST、THENなどの命令で指定された、ラインやラベルが存在しない。 ●プログラム入力時、ラインナンバーが65279を越えている。
5	深みエラー ●GOSUB(10段)、FOR−NEXT(5段)での段数オーバー ●RETURN文に対するGOSUB文がない。 NEXT文に対するFOR文がない READ文に対するDATA文がない ●データバッファ（8段）をオーバー ファンクションバッファ（16段）をオーバー 文字記憶エリア（80文字）をオーバー

エラーコード	エ　ラ　ー　内　容
6	容量オーバー ●プログラム及び変数が大きすぎて、プログラム・データエリアの容量を越えた
7	フォーマットエラー ●PRINT、PAUSE、LPRINT命令において数値データを表示あるいは印字する際、USING命令で指定されたフォーマットで表示あるいは印字できない。
8	入出力エラー プリンタ、テープレコーダーの入出力におけるエラー
9	その他のエラー ●数値変数に文字を代入しようとした。また逆に文字変数に数値を代入しようとした、など変数名の不適合 例　A$=123 ENTER 　　B$=A*2 ENTER ●パスワードの不一致など

入力ミスに関して

プログラムを実行したとき、プログラムの入力ミスによりエラーになることがあります。この場合、次のことに留意してください。

⊙入力ミスを訂正するような場合

（例）LPRINTをKPRINTと入力していたとき

```
                        10:K PRINT A$
          [►]           10 K PRINT A$
L         [ENTER]       10:L PRINT A$
                           └スペースがある
```

このような方法での修正では、計算機には1つの命令として判断されません。

このような場合はKPRINTを削除して、改めてLPRINTと入力してください。

```
                                          10:K PRINT A$
                                   [►]    10 K PRINT A$
[SHIFT] [◄](DEL) [SHIFT] [◄](DEL)         10 A$
[SHIFT] [►](INS) ‥[SHIFT] [►](INS)        10 ▯▯▯▯▯▯ A$
LPRINT                         [ENTER]    10:LPRINT A$
                                             └1つの命令と判断されたときは
                                              スペースは入らない。
```

なお、命令が正しく入力されているかどうかはカーソルでチェックすることができます。

（正しい入力の場合）

```
      10:RADIAN
[►]   10 RADIAN
[►]   10 RADIAN _
```

（まちがった入力の場合）

```
      10:RAD AN
[►]   10 RAD AN
[►]   10 RAD AN
[►]   10 RAD AN
```

キャラクタ・コード表

下位↓	上位→ 16進	0 0	16 1	32 2	48 3	64 4	80 5	96 6	112 7	128 8	………	224 E	240 F
0	0				0	@	P						
1	1			!	1	A	Q						
2	2			〃	2	B	R						
3	3			#	3	C	S						
4	4			$	4	D	T						
5	5			%	5	E	U						
6	6			&	6	F	V						
7	7			'	7	G	W						
8	8			(	8	H	X						
9	9			)	9	I	Y						
10	A			*	:	J	Z						
11	B			+	;	K	[						π
12	C			,	<	L	¥						√
13	D			−	=	M	]						
14	E			.	>	N	∧						
15	F			／	?	O	_						

キャラクタコードは次のように表わします。

例　"＊" のコード　　　　"√‾" のコード

16進数　&2A	16進数　&FC
10進数　42（32+10）	10進数　252（240+12）

注）表中のコード0（&00）のキャラクタは（ヌル）NULを示します。
また、コード32（&20）のキャラクタはスペースです。

注）本機は斜線部分にあたるコード、キャラクタは持っていません。
CHR$命令により、本機が持っていないキャラクタコードを指定しますと、実行時エラーになります。

BASICモードで使える関数・命令（省略形）一覧

1．基本関数

関数	省略形	備考
ABS	AB.	絶対値(アブソリュート)
ACS	AC.	$\cos^{-1}$
AHC		$\cosh^{-1}$
AHS	AH.	$\sinh^{-1}$
AHT		$\tanh^{-1}$
ASN	AS.	$\sin^{-1}$
ATN	AT.	$\tan^{-1}$
COS		cos
CUR	CU.	$\sqrt[3]{\ }$
DECI		16進→10進変換
DEG		度分秒→10進数度変換
DMS		10進数度→度分秒変換
EXP	EX.	e^x
FACT	FA.	n!
HCS	HC.	cosh
HEX		10進→16進変換
HSN	HS.	sinh
HTN	HT.	tanh
INT		整数化(インテジャー)
LN		ln　$\log_e X$
LOG	LO.	$\log_{10} X$
PI　(π)		円周率を指定(π≒3.141592654)
POL	PO.	直交座標→極座標変換
RCP	RC.	$\frac{1}{x}$　逆数
REC		極座標→直交座標変換
ROT	RO.	$\sqrt[x]{y}$　べき乗根
SGN	SG.	符号関数(シグナム)
SIN	SI.	sin
SQR ($\sqrt{\ }$)	SQ.	開平
SQU		x^2
TAN	TA.	tan
TEN	TE.	10^x
^		y^x　べき乗

2. BASIC命令

（※……プログラムで実行可能な命令
★……マニュアル操作で実行可能な命令）

	命　令	省略形	一般形	備　考
※	AREAD （オート リード）	A. 〜 AREA.	AREAD　変数	定義付けキーによるプログラム実行開始時に、表示内容を指定された変数に入れる。
※ ★	BEEP （ビープ）	B. 〜 BEE.	BEEP　式	音発生命令
※ ★	CLEAR （クリア）	CL. 〜 CLEA.	CLEAR	データ（変数）をすべて消去する命令。
※	DATA （データ）	DA. DAT.	DATA　式，式，…	データ指定 READ命令に続く変数に与えるデータを指定。
※ ★	DEGREE （ディグリー）	DE. 〜 DEGRE.	DEGREE	角度単位指定 ″度″単位指定（°）
※ ★	DIM （ディメン ジョン）	D. DI.	(1)DIM配列名(式) (2)DIM配列名(式)＊式$_3$ (3)DIM配列名(式$_1$，式$_2$) (4)DIM配列名(式$_1$，式$_2$)＊式$_3$ 配列名：B、C$、D$など (　　)：配列の大きさ、次元指定 式$_3$　：桁数指定（文字変数のみ）	変数の定義 配列変数の配列名大きさなどを定義し、プログラム・データエリアに確保する。
※	END （エンド）	E. EN.	END	プログラム終了命令

命　令	省略形	一　般　形	備　考
※ FOR (フォー) TO (トゥー) STEP (ステップ)	F. FO. STE.	(1)FOR　数値変数＝式$_1$ TO　式$_2$ (2)FOR　数値変数＝式$_1$ TO 式$_2$　STEP式$_3$ 式$_1$：初期値、式$_2$：最終値 式$_3$：増(減)分	FOR−NEXT ループの始まり NEXT命令と対 で使用
※ GOSUB (ゴーサブ)	GOS. GOSU.	GOSUB {式 / "文字" / 文字変数} ●指定されたラインあるいはラベルへ実行を移しそこからのプログラムをサブルーチンとして実行。 RETURN命令と対で使用。	サブルーチンジャンプ命令
※★ GOTO (ゴートゥ)	G. ∫ GOT.	GOTO {式 / "文字" / 文字変数} ●指定されたラインあるいはラベルへ実行を移す ●指定されたラインからプログラムを実行する。	ジャンプ命令 (プログラム実行開始命令)
※★ GRAD (グラード)	GR. GRA.	GRAD	角度単位指定 "グラード"単位指定(^{g})
※ IF (イフ)		IF {条件式 / 式} 実行文	判断命令 与えられた条件を判断し、実行を次の実行文か、あるいは次のラインへ移します。
※ INPUT (インプット)	I. ∫ INPU.	(1) INPUT変数,変数,… (2) INPUT"文字",変数, "文字",変数,… (3) INPUT"文字";変数, "文字";変数,…	入力命令 データを入力
※ LET (レット)	LE.	(1)LET数値変数＝式 (2)LET文字変数＝"文字" (3)LET文字変数＝文字変数	LETはIF文に続く場合以外は省略可

	命　令	省略形	一般形	備　考
※ ★	RADIAN (ラディアン)	RAD. 〜 RADIA.	RADIAN	角度単位指定 "ラディアン"単位指定〔rad〕
※	READ (リード)	REA.	READ変数，変数，…	データの読み込み命令 DATA命令に続いて指定されているデータを指定の変数に入れる。
※	REM (リマーク)		REM注釈	プログラムの中に非実行文（注釈）を指定するための命令。
※	RESTORE (リストア)	RES. 〜 RESTOR.	(1)RESTORE {式／"文字"／文字変数} (2)RESTORE	READ命令により読み込まれるデータの順番を変える命令。
※	RETURN (リターン)	RE. 〜 RETUR.	RETURN	復帰命令 サブルーチンの復帰命令で、GOSUBの次のステートメントへ実行が復帰します。
※	STOP (ストップ)	S. 〜 STO.	STOP	プログラムの実行を停止する命令。
※	THEN (ゼン)	T. 〜 THE.	IF文における実行文として定義します。 THEN {式／"文字"／文字変数} THEN¦BASIC命令¦	IF文における実行文として定義することのみ可能。

命　令	省略形	一般形	備　考
※ LPRINT (ラインプリント)	LP. ∫ LPRIN.	(1)LPRINT {式 / "文字" / 文字変数} (2)LPRINT {式 / "文字" / 文字変数} , {式 / "文字" / 文字変数} (3)LPRINT {式 / "文字" / 文字変数} ; {式 / "文字" / 文字変数} ……	出力命令 指定された内容を印字します。
※ NEXT (ネクスト)	N. ∫ NEX.	NEXT　数値変換 数値変数はFOR命令に続く数値変数と一致していなければならない。	FOR−NEXTループの末端を示します。FOR文と対で使用。
※ ON GOSUB (オンゴーサブ)	ONGOS. O. GOS. O.GOSU.	ON　式　GOSUB式$_1$，式$_2$，式$_3$，…	サブルーチンジャンプ命令。 ONに続く式の値によりジャンプ先を指定。
※ ON GOTO (オンゴートゥー)	ONG. O. G. ∫ O. GOT.	ON　式　GOTO式$_1$，式$_2$，式$_3$，…	ジャンプ命令 ONに続く式の値によりジャンプ先を指定する。
※ PAUSE (ポーズ)	PAU. PAUS.	一般形はPRINT命令と同じ形をとります。	出力命令 指定された内容を約0.85秒表示したのちプログラムを実行。

	命　令	省略形	一　般　形	備　考
※	PRINT (プリント)	P. ～ PRIN.	(1)PRINT {式 / "文字" / 文字変数} (2)PRINT {式 / "文字" / 文字変数} , {式 / "文字" / 文字変数} (3)PRINT {式 / "文字" / 文字変数} ; {式 / "文字" / 文字変数} …	出力命令 指定された内容を表示します。
※ ★			PRINT=LPRINT PRINT=PRINT	PRINT命令をLPRINT命令に指定 PRINT→LPRINTを解除
※ ★	USING (ユージング)	U. ～ USIN.	(1)USING "♯…♯. ♯…♯へ" (2)USING "& &……& & &" (3)PRINT USING "フォーマット" ; ……… (4)USING (5)PRINT USING;…	フォーマット指定命令 数値データ、文字データの出力フォーマット指定 (4)(5) フォーマット指定解除
※ ★	WAIT (ウエイト)	W. ～ WAI.	(1)WAIT 式 (式の値は0～65535) (2)WAIT	表示時間指定 PRINT命令によるプログラム停止時間を指定 指定を解除 (停止時間無限)

3. コマンド

	命　令	省略形	一　般　形	備　考
★	CONT （コンティニュー）	C. 〜 CON.	CONT	一時停止しているプログラムの実行を再開する命令 RUNモードで有効。
★	LIST （リスト）	L. 〜 LIS.	(1)LIST (2)LIST { 式 / "文字" / 文字変数 }	プログラムリストをおこなう命令。 PROモードで有効。
★	LLIST （ラインリスト）	LL. 〜 LLIS.	(1)LLIST (2)LLIST 式 (3)LLIST式$_1$，式$_2$ 式$_1$あるいは式$_2$を省略した形	リスト命令 プリンタによりプログラムのリストアウトをおこなう命令。
※ ★	MEM （メム）	M. ME.	MEM	プログラム・データエリアの中で空いている部分のバイト数を表示。
★	NEW （ニュー）		NEW	PROモードでプログラムおよび全変数をクリア。
★	PASS （パス）	PA. PAS.	PASS "文字"	パスワードの設定解除をする命令。
★	RUN （ラン）	R. RU.	(1)RUN (2)RUN 式 (3)RUN { "文字" / 文字変数 }	プログラム実行開始命令。 RUNモードで有効。
※ ★	TR OFF （トレースオフ）	TROF.	TR OFF	デバッグをおこなうモードを解除。
※ ★	TR ON （トレースオン）	TR. TRO.	TR ON	デバッグをおこなうモードを指定。

4．BASIC関数・その他

	命　令	省略形	一　般　形	備　考
※ ★	AND （アンド）	AN．	式　AND　式	論理積
※ ★	ASC （アスキー）		ASC｛"文字" 文字変数｝	文字などのキャラクタをアスキーコードに変換。
※ ★	CHR$ （キャラクタドル）	CH． CHR．	CHR$　式	アスキーコードを文字などのキャラクタに変換。
※ ★	INKEY$ （インキードル）	INK． ～ INKEY．	文字変数＝INKEY$	INKEY$命令実行時にキーが押されていればその内容を読み込む。
※ ★	LEFT$ （レフトドル）	LEF． LEFT．	LEFT$（文字変数，式） LEFT$（"文字"，式）	指定された文字列の左側から、指定された桁数の文字を取り出す。
※ ★	LEN （レングス）		LEN｛"文字" 文字変数｝	指定された文字列の文字数を求めます。
※ ★	MID$ （ミッドドル）	MI． MID．	MID$（文字変数，式$_1$，式$_2$） MID$（"文字"，式$_1$，式$_2$）	指定された文字列の中間の文字を取り出す。
※ ★	NOT （ノット）	NO．	NOT　式	否　定
※ ★	OR （オア）		式　OR　式	論理和
※ ★	RIGHT$ （ライトドル）	RI． ～ RIGHT．	RIGHT$（文字変数，式） RIGHT$（"文字"，式）	指定された文字列の右側から、指定された桁数の文字を取り出す。

	命　令	省略形	一般形	備　考
※ ★	RND （ランダム）	RN.	RND 式	乱数を得る命令。
※ ★	STR$ （ストリングドル）	STR.	STR$ 式 （A$=STR$ A）	数値を文字列に変換。
※ ★	VAL （バリュー）	V. VA.	VAL { "文字" / 文字変数 }	文字列を数値に変換。

5．テープレコーダーに関する命令

	命　令	省略形	一　般　形	備　考
★	CLOAD （カセットロード）	CLO. CLOA.	(1)CLOAD (2)CLOAD"ファイル名"	転送命令 プログラムをテープから計算機に転送する命令。
★	CLOAD? （カセットロード?）	CLO. ? CLOA. ?	(1)CLOAD? (2)CLOAD?"ファイル名"	照合命令 計算機内のプログラムとテープに記録されている内容との照合をおこなう命令。
★	CSAVE （カセットセイブ）	CS. ∫ CSAV.	(1)CSAVE (2)CSAVE"ファイル名" (3)CSAVE "ファイル名"，"パスワード" (4)CSAVE，"パスワード"	記録命令 プログラムをテープに記録する命令 (3)(4) プログラムを秘密化する。
※ ★	INPUT# （インプットクロスハッチ）	I. # ∫ INPU. #	(1)INPUT#変数名，変数名，… (2)INPUT#"ファイル名"；変数名，変数名，……	データ転送命令 テープに記録されているデータを指定された変数に転送する命令。
※ ★	PRINT# （プリントクロスハッチ）	P. # ∫ PRIN. #	一般形はINPUT#命令と同じ形となります。	データ記録命令 計算機に記憶されているデータをテープに記録する命令。

索　引

(ページ)

ABS……71
ACS……70
AHC……70
AHS……70
AHT……70
AND……139
AREAD……134
ASC……147
ASCII……147
ASN……70
ATN……70
BASIC……5
BASICモード……14
BEEP……137
CALモード……14
CHR$……147
CLEAR……118
CLOAD……158
CLOAD?……159
CONT……133
COS……70
CSAVE……156
CUR……70
DATA……130
DECI……71
DEG……71
DEGREE……72
DIM……110
DMS……71
END……100
EXP……70
FACT……70
FOR……104
GOSUB……122
GOTO……100

(ページ)

GRAD……72
HCS……70
HEX……71
HSN……70
HTN……70
IF……103
INKEY$……136
INPUT……95
INPUT#……162
INT……71
LEFT$……143
LEN……145
LET……100
LIST……86
LLIST……155
LN……70
LOG……70
LPRINT……152
MEM……138
MID$……143
NEXT……104
NOT……139
ON GOSUB……132
ON GOTO……132
OR……139
PASS……149
PAUSE……128
PI……70
POL……71
PRINT……98
PRINT#……160
PROモード……14
RADIAN……72
RAM……5
RANDOM……135

(ページ)

RCP……70
READ……130
REC……71
REM……133
RESTORE……131
RETURN……122
RIGHT$……143
RND……135
ROM……5
ROT……71
RUN……90
RUNモード……14
SGN……71
SIN……70
SQR……70
SQU……70
STEP……104
STOP……133
STR$……146
TAN……70
TEN……70
THEN……103
TO……104
USING……125
VAL……146
WAIT……128

(ページ)
アスキーコード……147
一次元配列……111
一変数統計計算……44
エラー……65
エラーコード……240
エラーメッセージ……91
円周率……70
音発生機能……137
カーソル……13
階乗……29,70
角度指定……18
角度変換……30
仮数部……65
カッコ計算……39
きざみ値……104
キャラクタコード……243
記録……156
逆三角関数……26,70
逆数計算……29,70
逆双曲線関数……27
行……83
行番号……83
くり返し……104
グラード……18,72
固定小数点方式……17,65
固定変数……108
最終値……104
サブルーチン……122
三角関数……25,70
座標変換……31
式……99
指数関数……28,70
指数部……65
指数方式……65
終了文……100
出力命令……98
照合……159

(ページ)
初期値……104
2乗計算……28,70
実行モード……14
10進→16進変換……33,71
ジャンプ命令……100
16進→10進変換……33,71
条件式……104
数値変数……68
スキップ……97
ステートメント……84
ステップ数……105
整数化……71
絶対値……71
相関係数……47
双曲線関数……27,70
添字……110
対数関数……27,70
単純変数……109
代入文……100
ダイレクトアンサー機能……76
注釈……133
テーブル変数……113
定義……110
定義付け……121
定数計算……37
転送……158
データ専用エリア……117
データバッファ……78
データメモリー……68
ディグリー……18
電卓モード……14
電卓用メモリー……23
統計計算……43
度……72
流れ図……82
二次元配列変数……113
二変数統計計算……46

（ページ）

入力ガイダンス……………………………96
入力命令…………………………………95
配列変数…………………………………110
配列要素…………………………………110
判断………………………………………103
バイト……………………………………5
パーセント計算…………………………36
パスワード………………………………149
否定………………………………………139
秘密プログラム…………………………149
ファイル名………………………………156
ファンクションバッファ………………78
フォーマット指定………………………125
符号関数…………………………………71
浮動小数点方式…………………………17,65
フローチャート…………………………82
フローチャート記号……………………94
ブレーク…………………………………133
ブレークメッセージ……………………133
文…………………………………………84
プレイバック機能………………………61
プログラム・データエリア……………117
プログラムモード………………………14
プロンプト記号…………………………91
平方根……………………………………28,70
変化率計算………………………………36
変数………………………………………68
べき乗……………………………………31,71
べき乗根…………………………………31,71
マニュアル計算…………………………56
マルチステートメント…………………84
メインルーチン…………………………122
モード……………………………………14
文字変数…………………………………68
優先順位…………………………………41,77
ライン……………………………………83
ラインナンバー…………………………83

（ページ）

ラストアンサー機能……………………63
ラディアン………………………………18,72
ラベル……………………………………101
乱数………………………………………135
リスト変数………………………………111
立方根……………………………………28
連続計算機能……………………………62
論理積……………………………………139
論理和……………………………………139

保　証　書（保証規定）

本書は、本書記載内容で無料修理をさせて頂くことをお約束するものです。

保証期間中に故障が発生した場合は、製品と本書をご持参、ご提示の上、お買上げの販売店にご依頼ください。

お買上げ年月日、販売店名など記入もれがありますと無効となります。必ずご確認いただき、記入のない場合はお買上げの販売店にお申し出ください。

ご転居・ご贈答品などでお買上げ販売店に修理をご依頼できない場合は、製品に同梱しております「シャープサービス・お客様ご相談窓口一覧表」をご覧の上、もよりのサービス会社へご持参ください。

本書は再発行いたしません。たいせつに保管してください。

〈無料修理規定〉

1．取扱説明書・本体注意ラベルなどの注意書にしたがった正常な使用状態で、保証期間内に故障した場合には、お買上げ販売店、または当社サービス会社が無料修理いたします。

2．保証期間内でも次の場合には有料修理となります。

（イ）本書のご提示がない場合。

（ロ）本書にお買上げ年月日・お客様名・販売店名の記入がない場合、または字句を書き換えられた場合。

（ハ）使用上の誤り、または不当な修理や改造による故障・損傷。

（ニ）お買上げ後に落とされた場合などによる故障・損傷。

（ホ）火災・公害および地震・風水害その他天災地変など、外部に要因がある故障・損傷。

（ヘ）電池の液もれによる故障・損傷。

（ト）消耗品（リチウム電池）が損耗し取り替えを要する場合。

3．本書は日本国内においてのみ有効です。

(THIS WARRANTY CARD IS ONRY VALID FOR SERVICE IN JAPAN.)

★この保証書は本書に明示した期間・条件のもとにおいて無料修理をお約束するものです。したがいましてこの保証書によってお客様の法律上の権利を制限するものではありませんので、保証期間経過後の修理などにつきましておわかりにならない場合はお買上げの販売店、またはシャープお客様ご相談窓口にお問い合せください。

修理メモ

シャープ計算機保証書
(WARRANTY CARD)

品　　名　　計　算　機

形　　名　　PC-1401

保証期間 (VALIDITY)　　お買上げ日より本体1年間 (FULL 1 YEAR AFTER PURCHASE)

お買上げ日 (PURCHASE)　　昭和　　年　　月　　日

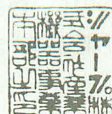
お客様	ご芳名	様
	ご住所	〒
	電話番号	(　　　)　　　－

取扱販売店名・住所・電話番号

日本大学生産工学部実籾校舎購買部
合資会社 大藪隆司商店
千葉県習志野市新栄2-11-1
〒275 電話0474(73)1211内線536

印

シャープ株式会社
産業機器事業本部
〒639-11　奈良県大和郡山市美濃庄町492
電話　(07435)3－5521番



シャープ株式会社

本社	〒545 大阪市阿倍野区長池町22－22	電話（06）621-1221（大代表）
産業機器事業本部	〒639-11 奈良県大和郡山市美濃庄町492	電話（07435）3－5521（大代表）
国内産機営業本部	〒545 大阪市阿倍野区長池町22－22	電話（06）621-1221（大代表）

プログラムや製品、機能についてのお問い合わせは下記へ………

シャープビジネス株式会社

〒545 大阪市阿倍野区長池町22－22 電話 (06)621－1221（大代表）

札幌支店	〒063	札幌市西区二十四軒1条7丁目3番17号	電話(011)641－3631
仙台支店	〒983	仙台市6丁目字本新田東2番の1	電話(0222)88－9151
宇都宮支店	〒320	宇都宮市不動前4丁目2番41号	電話(0286)37－3576
東京支店	〒130	東京都墨田区石原2丁目12番3号	電話 (03)625－5111
千葉支店	〒280	千葉市南町1丁目5番20号	電話(0472)63－4043
横浜支店	〒235	横浜市磯子区中原1丁目2番23号	電話(045)751－3215
埼玉支店	〒330	埼玉県大宮市宮原町2丁目107番地2号	電話(0486)63－5159
新潟支店	〒950	新潟市上所中1丁目7番21号	電話(0252)83－1795
長野支店	〒380	長野市中御所2丁目11番3号	電話(0262)28－4618
名古屋支店	〒454	名古屋市中川区山王3丁目5番5号	電話(052)332－2631
金沢支店	〒921	石川県石川郡野々市町御経塚1096番1号	電話(0762)49－1240
大阪支店	〒556	大阪市浪速区恵美須西1丁目2番9号	電話 (06)643－3021（代表）
京都支店	〒601	京都市南区上鳥羽菅田町48番地	電話(075)661－7335
神戸支店	〒658	神戸市東灘区魚崎北町1丁目6番地18号	電話(078)452－8531
広島支店	〒731-01	広島市安佐南区祇園町大字西原2249番地の1	電話(082)874－4925
高松支店	〒760	香川県高松市木太町1861番地の3	電話(0878)33－4255
福岡支店	〒816	福岡市博多区井相田2丁目12番地の1	電話(092)572－2611
沖縄支店	〒900	沖縄県那覇市曙2丁目10番地の1	電話(0988)61－7360（代表）

修理などアフターサービスについてのお問い合わせは下記へ……

シャープお客様ご相談窓口（技術センター）

札幌 技術センター	〒(063)	札幌市西区二十四軒1条7丁目3番17号	電話(011)641－0751
仙台 技術センター	〒(983)	仙台市6丁目字本新田東2番地の1	電話(0222)88－9161
宇都宮技術センター	〒(320)	宇都宮市不動前4丁目2番41号	電話(0286)34－0256
新潟 技術センター	〒(950)	新潟市上所中1丁目7番21号	電話(0252)84－6023
東京第1技術センター	〒(130)	東京都墨田区石原2丁目12番3号	電話 (03)624－7475
東京第2技術センター	〒(164)	東京都中野区南台3丁目45番13号	電話 (03)382－9161
東京第3技術センター	〒(143)	東京都大田区南馬込1丁目5番15号	電話 (03)777－8851
大宮 技術センター	〒(330)	大宮市宮原町2丁目107番2号	電話(0486)66－7220
横浜 技術センター	〒(235)	横浜市磯子区中原1丁目2番23号	電話(045)753－9583
名古屋技術センター	〒(454)	名古屋市中川区山王3丁目5番5号	電話(052)332－2671
金沢 技術センター	〒(921)	石川県石川郡野々市町字御経塚町1096の1	電話(0762)49－9033
大阪第1技術センター	〒(556)	大阪市浪速区恵美須西1丁目2番9号	電話 (06)644－1191
大阪第2技術センター	〒(567)	茨木市鮎川5丁目15番3号	電話(0726)34－4683
神戸 技術センター	〒(658)	神戸市東灘区魚崎北町1丁目6番18号	電話(078)452－1762
京都 技術センター	〒(601)	京都市南区上鳥羽菅田町48番地	電話(075)681－9551
高松 技術センター	〒(761-01)	高松市高松町3009番地の4	電話(0878)43－3711
岡山 技術センター	〒(700)	岡山市米倉字東ノ町66の2	電話(0862)43－5660
広島 技術センター	〒(731-01)	広島市安佐南区祇園町大字西原2249番地の1	電話(082)874－6100
福岡 技術センター	〒(816)	福岡市博多区井相田2丁目12番1号	電話(092)572－2617
鹿児島技術センター	〒(890)	鹿児島市鴨池新町12番1号	電話(0992)59－0628

5A7.5T(TINSJ4360CCZZ)

SHARP

全国をくまなくめぐるシャープのサービス網

このたびはシャープ製品をお買上げいただきましてありがとうございます。
シャープ製品のアフターサービスはお買上げ販売店、または下記シャープシステムサービス㈱がさせていただきます。
なお、アフターサービスについておわかりにならないことは、もよりの技術センターへお問合せください。その他ご相談、お問合せは各地のお客様ご相談窓口をご利用ください。
この一覧表は取扱説明書、保証書などと共に保存してください。

シャープサービス・お客様ご相談窓口一覧表

都道府県名	サービス拠点名	電話番号	郵便番号	所在地
北海道	**札幌技術センター**	(011)641-0751	063	札幌市西区二十四軒1条7丁目3番17号
	札幌技術センター 帯広	(0155)35-4068	080	帯広市西11条南1丁目12番地1
	札幌技術センター 旭川	(0166)22-8284	070	旭川市一条通4丁目左10号
	札幌技術センター 函館	(0138)52-5190	040	函館市五稜郭町31番17号
	札幌技術センター 釧路	(0154)24-8191	085	釧路市光陽町8番地の13
	札幌技術センター 北見	(0157)36-6814	090	北見市三輪町435番地
	札幌技術センター 苫小牧	(0144)74-8070	053	苫小牧市有明町1丁目5番6号
宮城	**仙台技術センター**	(0222)88-9161	983	仙台市六丁目字本新田東2番地の1
福島	仙台技術センター 郡山	(0249)46-0196	969-05	郡山市安積町荒井字方八丁33番地の1
	仙台技術センター いわき	(0246)28-2487	970	いわき市自由ケ丘37番地の10
岩手	仙台技術センター 盛岡	(0196)38-6085	028-36	岩手県紫波郡矢巾町流通センター南3丁目1番1号
	仙台技術センター 水沢	(01972)4-1710	023	水沢市卸町3番10号
秋田	仙台技術センター 秋田	(0188)65-1258	010	秋田市川尻町字大川反170番地の56
青森	仙台技術センター 青森	(0177)38-7778	030-01	青森市大字野尻字今田94-41
	仙台技術センター 八戸	(0178)45-2631	031	八戸市小中野2丁目8番16号
山形	仙台技術センター 山形	(0236)31-6634	990-23	山形市蔵王飯田227番地1
	シャープ事務機山形販売㈱	(0236)33-3215	990-22	山形市流通センター1丁目3番地の2
栃木	**宇都宮技術センター**	(0286)34-0256	320	宇都宮市不動前4丁目2番41号
	宇都宮技術センター 小山	(0285)25-6532	323	小山市城東1丁目17番10号
群馬	宇都宮技術センター 前橋	(0272)23-1311	371	前橋市表町1丁目4の1
	宇都宮技術センター 太田	(0276)46-4110	373	太田市東長岡1788
茨城	宇都宮技術センター 水戸	(0292)43-0909	310	水戸市千波町1963番地
	宇都宮技術センター 土浦	(0298)23-7673	300	土浦市真鍋町6丁目20班2255番地
新潟	**新潟技術センター**	(0252)84-6023	950	新潟市上所中1丁目7番21号
	新潟技術センター 長岡	(0258)23-1850	940-11	長岡市摂田屋町字嶺2600番
	新潟技術センター 上越	(0255)22-3360	943	上越市大字土橋字大坪1905番地の1
長野	新潟技術センター 松本	(0263)27-1636~7	399-65	松本市芳野8番14号
	新潟技術センター 長野	(0262)26-6208	380	長野市中御所2丁目11番3号
	新潟技術センター 上田	(0268)27-9395	386	上田市住吉67-16
東京	第1技術部・第2技術部	(03) 260-1161(大代表)	162	新宿区市谷八幡町8番地シャープ東京ビル内
	東京第1技術センター	(03) 624-7475	130	墨田区石原2丁目12番3号シャープ江東ビル内
	東京第1技術センター 中央	(03) 436-0814	105	港区虎ノ門5丁目1番4号
	東京第1技術センター 城東	(03) 606-4330	120	足立区綾瀬7丁目22番11号
	東京第2技術センター	(03) 382-9161	164	中野区南台3丁目45番13号 シャープ城西ビル内
	東京第2技術センター 城北	(03) 973-5120	174	板橋区東新町1丁目33番11号
	東京第2技術センター 武蔵野	(03) 326-2181	157	世田谷区給田町5丁目19-1
	東京第2技術センター 日野	(0425)85-1656	191	日野市日野台5丁目5番4号
	東京第3技術センター	(03) 777-8851	143	大田区南馬込1丁目5番15号シャープ城南ビル内
埼玉	**大宮技術センター**	(0486)66-7220	330	大宮市宮原町2丁目107番2号
	大宮技術センター 熊谷	(0485)54-5140	361	行田市門井町2丁目5番地
	大宮技術センター 戸田	(0484)43-2788	335	戸田市上戸田3丁目26番6号
千葉	東京第1技術センター 千葉	(0472)65-6110	280	千葉市南町1丁目5番20号
	東京第1技術センター 松戸	(0473)68-5166	271	松戸市稔台295番1
	東京第1技術センター 船橋	(0474)23-7170	273	船橋市市場4丁目16番2号
山梨	東京第2技術センター 甲府	(0552)28-1320	400	甲府市富竹2丁目1番17号
神奈川	**横浜技術センター**	(045)753-9583	235	横浜市磯子区中原1丁目2番23号
	横浜技術センター 湘南	(0463)55-7600	254	平塚市田村字満向1381
	横浜技術センター 相模原	(0427)59-4398	229	相模原市渕野辺3-11
静岡	横浜技術センター 静岡	(0542)83-9497	422	静岡市曲金6丁目8番44号
	横浜技術センター 沼津	(0559)24-1028	410	沼津市宮前町11-4
	名古屋技術センター 浜松	(0534)65-0735	430	浜松市植松町1476-2
愛知	**名古屋技術センター**	(052)332-2671	454	名古屋市中川区山王3丁目5番5号シャープ名古屋ビル内
	名古屋技術センター 豊橋	(0532)54-1830	440	豊橋市下地町橋口17の1
	名古屋技術センター 半田	(0569)21-1369	475	半田市住吉町2-109
	名古屋技術センター 岡崎	(0564)25-0611	444	岡崎市柿田町1番地21
岐阜	名古屋技術センター 岐阜	(0582)74-7996	500	岐阜市六条南3丁目12番9号
	名古屋技術センター 濃飛	(0574)25-8418	509-02	可児市土田字下切3832-1
三重	名古屋技術センター 津	(0592)31-1573	514-01	津市栗真町屋町字蒲池328番地
	名古屋技術センター 四日市	(0593)51-9434	510	四日市市鵜の森1丁目15番地9
石川	**金沢技術センター**	(0762)49-9033	921	石川県石川郡野々市町字御経塚町1096の1
富山	金沢技術センター 富山	(0764)51-3933	930	富山市金泉寺71番1号
福井	金沢技術センター 福井	(0776)53-6050	910	福井市北四ツ居町625番地
	シャープ事務機福井販売㈱	(0776)27-1800	910	福井市新田塚1丁目70番26号
大阪	第4技術部・第5技術部	(06) 621-1221(大代表)	545	大阪市阿倍野区長池町22番22号
	大阪第1技術センター	(06) 644-1191	556	大阪市浪速区恵美須西1丁目2番9号
	大阪第1技術センター 東大阪	(0729)93-2001	581	八尾市跡部北の町3丁目2番11号
	大阪第1技術センター 南大阪	(0722)45-5855	590	堺市老松町1丁目39番地
	大阪第2技術センター	(0726)34-4683	567	茨木市鮎川5丁目15番3号
奈良	大阪第1技術センター 奈良	(07435)3-2023~4	639-11	大和郡山市美濃庄町492番地
和歌山	大阪第1技術センター 和歌山	(0734)45-6298	641	和歌山市西小二里2丁目4番91号
	大阪第1技術センター 田辺	(0739)24-4092	646	田辺市稲成町441-1
兵庫	**神戸技術センター**	(078)452-1762	658	神戸市東灘区魚崎北町1丁目6番18号
	神戸技術センター 姫路	(0792)66-8295	671-22	姫路市青山字塔之内1006番の1
	神戸技術センター 阪神	(06) 422-6250	661	尼崎市猪名寺西田187番地
	神戸技術センター 豊岡	(07962)2-6174	668	豊岡市九日市上町下畑77の1
京都	**京都技術センター**	(075)681-9551	601	京都市南区上鳥羽菅田町48番地
	京都技術センター 舞鶴	(0773)76-6069	624	舞鶴市福来問屋町785-47
滋賀	京都技術センター 大津	(0775)43-2331	520-21	大津市栗林町11-35
	京都技術センター 彦根	(0749)23-8688	522	彦根市古沢町406番地の1
香川	**高松技術センター**	(0878)43-3711	761-01	高松市高松町3009番地4
愛媛	高松技術センター 松山	(0899)73-0121	791-41	松山市高岡町178番1
	高松技術センター 新居浜	(0897)43-8854	792	新居浜市国領1丁目甲1799番地の4
	高松技術センター 宇和島	(0895)24-1975	798	宇和島市中沢町1丁目1番20号
徳島	高松技術センター 徳島	(0886)25-8840	770	徳島市中常三島町3丁目11番地14
高知	高松技術センター 高知	(0888)83-7039	780	高知市高須字高須塩田西の丸960番1
岡山	**岡山技術センター**	(0862)43-5660	700	岡山市米倉字東ノ町66の2
	岡山技術センター 津山	(08682)2-6296	708	津山市沼6番地の8
鳥取	岡山技術センター 鳥取	(0857)26-4227	680	鳥取市青葉町2丁目204番地
広島	岡山技術センター 福山	(0849)52-0736	720	福山市津之郷町大字津之郷字上開地
	広島技術センター	(082)874-6100	731-01	広島市安佐南区祇園町大字西原2249の1
	広島技術センター 東広島	(0824)28-3061	739-01	東広島市八本松町大字米満字徳政36-3
山口	広島技術センター 徳山	(0834)32-3839	745	徳山市新地1番地
	広島技術センター 下関	(0832)56-6699	751	下関市秋根南町2丁目5番2号
	広島技術センター 小郡	(08397)2-4525	754	吉敷郡小郡町大字下郷2255-39
島根	広島技術センター 松江	(0852)21-6110	690	松江市西津田3丁目1番10号
福岡	**福岡技術センター**	(092)572-2617	816	福岡市博多区井相田2丁目12番地1号
	福岡技術センター 北九州	(093)592-6510	803	北九州市小倉北区大手町6番12号
	福岡技術センター 久留米	(0942)21-7858	830	久留米市上津町1682-2
	福岡技術センター 筑豊	(0948)24-9283	820	飯塚市大字川津字宮の前272-2
大分	福岡技術センター 大分	(0975)32-0518	870	大分市中春日町16番20号
長崎	福岡技術センター 長崎	(0958)45-8481	852	長崎市宝町6番11号
	福岡技術センター 佐世保	(0956)33-7288	857-11	佐世保市白岳町107-5
佐賀	福岡技術センター 佐賀	(0952)25-0983	840	佐賀市鍋島町大字八戸字五本松籠2043-2
鹿児島	**鹿児島技術センター**	(0992)59-0628	890	鹿児島市鴨池新町12番1号
宮崎	鹿児島技術センター 宮崎	(0985)28-8371	880	宮崎市原町4番12号
	鹿児島技術センター 延岡	(0982)21-7744	882	延岡市浜砂町1158番地
熊本	鹿児島技術センター 熊本	(0963)72-1251	862	熊本市新屋敷3丁目15番17号
沖縄	福岡技術センター 沖縄	(0988)66-5562	900	那覇市曙2丁目10番地の1

※ 上記の所在地、電話番号は変わることがありますので、その節はご容赦願います

84.11〈TLSTS1006CCZZ〉